NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION

MODELING THE TRANSFORMATIONAL COMMUNICATIONS SYSTEM URBAN LAND MOBILE SATELLITE CHANNEL

by

Timothy A. Gillespie

December 2007

Dissertation Supervisor: R. C. Robertson

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 2007	Dissertation

4. TITLE AND SUBTITLE: Modeling the Transformational Communications System Urban Land Mobile Satellite Channel	5. FUNDING NUMBERS
6. AUTHOR(S) Timothy T. A. Gillespie	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; distribution is unlimited	

13. ABSTRACT (maximum 200 words)

The success of the Transformational Communications System depends on providing high quality communications to the mobile user in any environment. A hurdle to this success is the urban Land Mobile Satellite Channel which experiences frequent and extended deep signal fades due to shadowing by man-made structures. The problem is exacerbated for the Communications-on-the-Move channel due to use of a directional antenna. This dissertation evaluates available empirical test data, develops an improved channel model and investigates mitigation strategies focused on improving channel throughput and latency. The improved channel model is developed using continuous transmission mixed-Gaussian hidden Markov models and Expectation-Maximization parameter estimation. The improved model provides a better match to empirical data than previous efforts. Two fade mitigation strategies are examined. The first strategy involves modification of packet routing protocols. A detailed, time-based network simulation is developed for performance assessment. Results show substantial reductions in packet latency under a broad range of conditions. The second mitigation strategy is based on fade prediction using real-time adaptive linear filters. Signal inputs from directional and non-directional antennas are investigated. Initial results indicate a requirement for a high temporal resolution input signal.

14. SUBJECT TERMS Land Mobile Satellite Channel, Transformational Communications System, Transformational Communications Architecture, Communications-on-the-move, Mobile Ad-hoc Network, Baum-Welch Algorithm, Least Resistance Routing, Neighbor protocol	15. NUMBER OF PAGES 225
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK6

Approved for public release; distribution is unlimited

MODELING THE TRANSFORMATIONAL COMMUNICATIONS SYSTEM URBAN LAND MOBILE SATELLITE CHANNEL

Timothy T. A. Gillespie
Major, United States Air Force
B.S., United States Air Force Academy, 1993
M.S., University of Colorado, 1994

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2007

Author: _____
Timothy T. A. Gillespie, Major United States Air Force

Approved by:

_____
R. Clark Robertson, PhD
Professor of Electrical and Computer Engineering
Dissertation Supervisor and Committee Chair

_____
Monique P. Fargues, PhD
Associate Professor of Electrical
and Computer Engineering

_____
Roberto Cristi, PhD
Professor of Electrical
and Computer Engineering

_____
Herschel H. Loomis, Jr., PhD
Professor of Electrical
and Computer Engineering

_____
Brij Agrawal, PhD
Professor of Mechanical
and Astronautical Engineering

Approved by: _____
Jeffrey B. Knorr, Chair, Department of Electrical and Computer Engineering

Approved by: _____
Julie Filizetti, Associate Provost for Academic Affairs

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The success of the Transformational Communications System depends on providing high quality communications to the mobile user in any environment. A hurdle to this success is the urban Land Mobile Satellite Channel which experiences frequent and extended deep signal fades due to shadowing by man-made structures. The problem is exacerbated for the Communications-on-the-Move channel due to use of a directional antenna. This dissertation evaluates available empirical test data, develops an improved channel model and investigates mitigation strategies focused on improving channel throughput and latency. The improved channel model is developed using continuous transmission mixed-Gaussian hidden Markov models and Expectation-Maximization parameter estimation. The improved model provides a better match to empirical data than previous efforts. Two fade mitigation strategies are examined. The first strategy involves modification of packet routing protocols. A detailed, time-based network simulation is developed for performance assessment. Results show substantial reductions in packet latency under a broad range of conditions. The second mitigation strategy is based on fade prediction using real-time adaptive linear filters. Signal inputs from directional and non-directional antennas are investigated. Initial results indicate a requirement for a high temporal resolution input signal.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The success of the future Transformational Communications System depends on providing high quality communications to the mobile user in any environment. A hurdle to this success is the Urban Land Mobile Satellite Channel (ULMSC) which experiences frequent and extended deep signal fades due to shadowing by man-made structures. The problem is exacerbated for the Communications-on-the-Move (COTM) channel due to use of a directional antenna. These signal fades are problematic for packet routing protocols and could hamper the network performance of the Communications-on-the-Move (COTM) link planned as part of the satellite-based Transformational Communication Architecture (TCA). This dissertation evaluates empirical data from Boston, MA and investigates mitigation strategies designed to improve COTM link throughput and latency.

In the time domain, the COTM channel is best modeled as a Markov generative process. The Baum-Welch parameter estimation algorithm is applied to empirical data collected in Boston, MA. The parameter estimation algorithm provides guidance for the development of a Markovian generative model to represent the ULMSC fade function in a larger time-based simulation. Three different continuous output transmission Markov models are presented and results are compared with discrete output Markov models. The continuous output models produce more accurate state durations, provide signal strength as a function of time and generally advance previous modeling efforts. The improved generative channel models also support the development of a larger time-based network simulation.

Local area, mobile ad-hoc networks provide an opportunity to mitigate the impacts of the urban signal fade phenomenon by circumventing the link outage. An integrated network simulation is developed for comparison against baseline performance. The simulation uses a Markov model to represent the urban shadowing and resulting channel fade. The Neighbor protocol results show substantial improvements in COTM packet latency and network throughput when the ad-hoc network is operating below capacity.

Finally, two approaches into fade prediction are introduced and limited results are provided. The first approach is based on adaptive linear filters using the signal strength received by a highly directional antenna as the filter input. The second approach is based on the combined line-of-sight and multi-path energy collected by an omni-directional system. Analytical results indicate that the abrupt signal strength transitions are inadequately predicted by linear filters and a high frequency input signal is important.

The study makes several recommendations for future work including additional collection of empirical data, improvements to the network simulation and further exploration of accurate fade prediction based on multipath energy collected by an omni-directional system.

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF SYMBOLS

$\mathbf{A}$	transition matrix
A	weighting coefficient of a mixture pdf
a_{ij}	probability of transitioning from state i to state j
a_i	Probability of Markov models starting in state i
α	link-dependent resistance weighting coefficient
B_1, B_2	weighting coefficients of a mixture pdf
$b_j(O_t)$	probability of generating output O_t while in state j
β	node-dependent resistance weighting coefficient
β_{GAL}	forgetting factor in GAL filter
c	speed of light
E[•]	expectation of [•]
e_1	number of hops erased from one packet transmission
e_2	number of symbol errors in one packet transmission
ε	tolerance
f(•)	probability density function for random variable (•)
h	probability of receiving an error while in the bad state (Gilbert model)
$I(A,B)$	link-dependent resistance between nodes A and B
$L(x\|y)$	likelihood of x given y
λ	set of parameters that uniquely define a Markov model
$N(\mu, \sigma)$	Normal or Gaussian distribution with mean μ and standard deviation σ
M	modulation order (MPSK)
m	number of mobile ad-hoc network nodes
n	number of TDMA time slots to send one packet
$\eta_{1,2}$	constants defining power law probability density function
$\mathbf{O}$	finite model output sequence of duration T
O_t	output of Markov model at time t
Pr(•),P(•)	probability of (•)
P	filter order (or memory) of linear filter
$\boldsymbol{\pi}$	Vector of all steady state probabilities

π_i	Steady state probability for state *i*
r	distance between mobile node and satellite
r_p	total combined packet rate at one mobile
r_{adhoc}	packet generation rate of local destination packets at one mobile
r_{sat}	packet generation rate of COTM packets at one mobile (uplink) or for one mobile (downlink)
S	number of states in the discrete-state Markov model
s	sequence of discrete states
s_t, $s(t)$	state at time *t*
σ_e^2	prediction error variance
T	total number of time steps in the finite sequence
t	current time step
U	Buffer utilization metric; fraction of buffer that is full
v(t)	velocity at time *t*

LIST OF ABBREVIATIONS AND ACRONYMS

AR	Autoregressive
ARQ	Automatic Repeat Request
BER	Bit error rate
BWA	Baum-Welch algorithm
CDF	Complementary distribution function
CDMA	code division multiplexing algorithm
COTM	Communications-on-the-move
CTS	Clear-to-send
DoD	Department of Defense
EE	errors and erasures
EEA	errors, erasures and acknowledgements
EM	Expectation-Maximization
FEC	Forward error correction
FHSS	Frequency-hop spread spectrum
GAL^2	Gradient Adaptive Lattice Ladder
GIG	Global Information Grid
HMM	hidden Markov model
HMMWV	High Mobility Multi-purpose Wheeled Vehicle
JTRS	Joint Tactical Radio System
LMRC	Land Mobile Radio Channel
LMSC	Land Mobile Satellite Channel
LOS	Line of sight
LPI	low-probability-of-intercept
LRR	Least resistance routing
m	meters
MANET	Mobile Ad-hoc Network
MOTM	MILSTAR-on-the-move
MPSK	*M*-ary phase-shift keying
Msym	Megasymbols
NLMS	Normalized least mean squares
OSI	Open Systems Interconnection
pdf	probability density function
pps	packets per second
ppts	packets per time-step
RTSM	Ray trace scatter model
RF	radio frequency
RLS	Recursive least squares
RTS	Request-to-send
Rx	Receive
SATCOM	satellite communications
SINCGARS	Single Channel Ground and Airborne Radio System
SNR	Signal-to-Noise Ratio

SOTM	Satcom-on-the-move
TCS	Transformational Communications System
TCA	Transformational Communications Architecture
TDMA	Time division multiplexing algorithm
TSAT	TCS Satellite
ULMSC	Urban Land Mobile Satellite Channel
VoIP	Voice over Internet Protocol
WDT	watchdog timer

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to all of the people who supported me in the completion of this dissertation. First and foremost is my dissertation supervisor and one of the best teachers I have ever met, Dr. Clark Robertson. Dr. Robertson taught me almost everything I know about communication theory and donated his time to motivate me through this arduous process. The other four members of the dissertation committee, Dr. Monique Fargues, Dr. Hershel Loomis, Dr. Roberto Cristi and Dr. Brij Agrawal, were also extremely helpful, both as strong instructors and supportive advisors. While not on the dissertation committee, Dr. Dave Jenn donated several hours of his time to teach me Urbana and help me contact the SAIC developer.

I also owe debts of gratitude to Dr. Jeff Schodorf and Dr. Scott Stadler of MIT/Lincoln Lab. In many ways, this dissertation stands on Dr. Schodorf's shoulders because it starts where his testing finishes. Dr. Scott Stadler provided the original topic suggestion and arranged my initial contact with Dr. Schodorf.

Another key source of support was the awesome team at Advanced Systems Division (ASD). Hardly a week went by without someone stopping to ask how the research was going, and when I did not understand something, they were happy to help me puzzle it out. Leading that motley band of crazy superheroes was the best boss a person could ever ask for, Dr. Laura Kennedy. She is an amazing role model and has taught me so much; it would have been even more amazing if I had been able to absorb it all.

I would like to offer special thanks to my two volunteer editors: Curtis Martin and Andy Forbes. Their constructive feedback was so important to this end product. Hopefully, I will be able to repay them someday.

And last, but certainly not least, I owe so much to my friends and family. For the past several years, they have silently offered their love and understanding, even when I could not spend time with them. And to Chris Miller, the best office mate and partner-in-crime, all I can say is "b/m."

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION AND BACKGROUND

The past several decades have seen significant changes in communications systems of all kinds. General changes have included rapid growth in the number and complexity of communication systems. Specific changes have included rapid growth in packet-based communications, spread spectrum applications, and systems with diversity. While some of this change has been fueled by terrestrial systems like cellular telephones, satellite-based communications have also seen improvements. These changes have raised the U.S. military's expectations regarding the quality, flexibility and robustness of future communication services. One area of increased expectation is developing a capability to maintain high data rate, secure communications with mobile military units in any environment. Initial research into developing this capability has found the urban environment to be the most challenging. An analysis of the urban satellite-to-mobile communications channel is presented and a solution to the associated fade problem is proposed and investigated in this dissertation.

A. INTRODUCTION TO THE TRANSFORMATIONAL COMMUNICATIONS SYSTEM

In 2001, the Department of Defense (DoD) began acquisition of a new ground- and satellite-based communications infrastructure [1]. This new program was named the Transformational Communications System (TCS) or Transformational Communications Architecture (TCA). The overall purpose of the TCA program is to design and build the next generation government communications infrastructure with increased capabilities, performance and interoperability. The TCA incorporates several planned satellite constellations and a new matching ground segment into a common Internet-like backbone, called the Global Information Grid (GIG) [1]. This architecture is the planned replacement of the hodge-podge, stove-piped systems in operation today (e.g., Defense Satellite Communication System, MILSTAR, Tracking and Data Relay Satellites, and UHF Follow-On). In addition to replacing current systems, the new Transformational

Communications System plans to add several new capabilities and improve performance of existing capabilities. Some specific examples of new or improved capabilities are discussed in Sections 1-3.

Figure 1. Artist's rendition of a TSAT [2].

The GIG and the TCA are built around a primary backbone of five geosynchronous satellites known as the Transformational Communications Satellites (TSATs). DoD acquisition of the TSATs is still in the early stage of Technology Development, but the System Design and Production phase is planned for 2008 [3]. The system's operational availability date has moved due to funding fluctuations, but the latest plans show a start in 2016-2020. The TSATs are planned to be connected by optical cross-links which provides the TCA with the backbone throughput necessary [1, 3]. The satellites communicate with their ground control station through a space-to-ground radio frequency (RF) link. This link is focused on collecting telemetry and issuing vehicle commands. Finally, each satellite has the primary task of supporting multiple RF links with various military users (i.e., naval ships, unmanned aerial vehicles,

mobile ground forces) [1, 3]. The TCA satellites establish these RF links with customers by providing several spot beams, each of which will illuminate an area the size of Nevada. Any customers inside the spot beam will be able to establish a link with the system. The uplink and downlink frequencies are expected to be 44 GHz and 20 GHz respectively, which are the same frequencies used by MILSTAR today [4, 5].

1. Packet-based Communications

The TCS program is currently planning to implement a packet-based communications architecture similar to TCP/IP [3, 5]. This is a significant shift from the more traditional point-to-point satellite communications architecture where each end involves a sizable antenna and equipment suite in a safe and stable location. The change reflects a general plan to increase the number of nodes/users using the network at any one time. The shift also reflects a change in the type of traffic that is expected. Users are no longer expected to be stable or consistent. Instead, users may access or drop off the network haphazardly, and their traffic patterns may be bursty and highly random. Instead of a large communications node, there will be numerous nodes in a variety of situations. Some users may be moving, others stationary. Some users may be in urban cities, others in open fields. Some users may be in combat situations, while others are in quiescent situations. The combination of all these changes has pushed the system towards a packet-based network approach. The network simulation presented in Chapters III and IV reflects the TCS move in this direction and represents a packet-based system.

2. Adaptive Communications Links

Another new capability proposed for TCS is the inclusion of adaptive communications links [4]. While this concept has been discussed for many years [6, 7], it has not been aggressively pursued until recently. The recent interest in adaptive communications has been driven by personal cellular communications and increased scarcity in available bandwidth, but the idea has also been discussed in reference to improving satellite communications performance [8, 9]. The basic concept behind an adaptive communications link is increased throughput when the channel is benign and

decreased throughput when the channel is experiencing increased attenuation due to weather or interference. The link throughput can be increased or decreased by changing several different parameters individually or in combination: signal power, data rate, and FEC code rate are the most popular suggestions. While this concept is not new, very few satellites have adaptive communication links. Instead, they have traditional "fixed" designs where all link parameters are calculated assuming worst-case channel conditions. In this simplistic approach, the system meets requirements when the channel is worst-case but does not increase throughput when the channel is benign. Non-adaptive designs have been popular because they are simple, inexpensive, and meet customer needs when bandwidth and power are readily available, but contemporary communication systems tend to have less power and bandwidth is becoming scarcer.

3. Communications-On-The-Move (COTM) Link

A third significant capability improvement associated with the TCA is extending the wideband data connection to smaller, mobile users in their operational environment. This capability is frequently referred to as Communications-on-the-Move (COTM) or SATCOM-on-the-Move (SOTM) [1, 5, 10, 11]. The COTM link is expected to provide Voice over Internet Protocol (VoIP) telephony, video streaming and internet-like web browsing to the individual vehicle in real-time. This capability is also expected to provide senior military leaders with near real-time knowledge of their troops' health, status and tactical capability. While the term COTM is relatively recent, the basic problem has traditionally been known as the Land Mobile Satellite Channel (LMSC). The two terms are used interchangeably in this dissertation, along with other synonymous terms like satellite communications (SATCOM) or satellite link. An introduction to LMSC research is provided in the next section.

Increased interest and government funding has fueled COTM related research and technology development in recent years. One early focus area was development of a small, agile and reliable antenna capable of maintaining the link. In 2003, Schodorf presented design and performance data for an early prototype [5]. More recently, L3

Communications, Perla Group International and other communications companies have started marketing production COTM antennas for the military, civil or commercial customer [10, 11]. To provide the required antenna gain in a small package, these antenna solutions are highly directional, which has an associated problem discussed below.

This dissertation focuses on a problem associated with the COTM link: long term signal fade in an urban environment. These fades are the result of vehicular motion amongst and between large buildings. This specific COTM problem is obviously limited to mobile land-based users. The user of interest is a relatively small vehicle (e.g., High Mobility Multi-purpose Wheeled Vehicle (HMMWV)) that may be in motion while attempting to maintain the communications link. The vehicle will be equipped with a directional antenna on the order of 12-16 inches in diameter [3, 5]. Figure 2 shows a prototype HMMWV test vehicle with the directional antenna installed.

Figure 2. HMMWV test vehicle with small, parabolic COTM antenna [5].

The beamwidth of this small antenna is expected to be around three degrees. This tight beamwidth has several significant impacts on performance. On one hand, it increases gain and throughput under nominal conditions. However, it also decreases the

amount of multi-path (scattered or reflected) signal energy that is received. Decreasing the amount of energy received from multi-path sources may be helpful when the line-of-sight (LOS) signal is present. But when the LOS signal is not available (e.g., blocked), the system collects almost no energy at all. The combination of small mobile user and directional antenna has led to a communications problem when the LOS signal is frequently blocked, like in an urban area with numerous buildings.

This dissertation is focused on understanding, modeling and mitigating the urban COTM channel fade problem. The DoD's development and deployment of the GIG and TCS are major acquisition efforts for the next decade. The U.S. Army has associated acquisition activities including the Future Combat System and the Joint Tactical Radio System that are counting on the COTM channel. The first step in understanding the COTM channel is to review the related research into the LMSC.

B. INTRODUCTION TO THE LAND MOBILE SATELLITE CHANNEL

As mentioned above, the COTM link planned for the upcoming TCA is a specific implementation of the Land Mobile Satellite Channel which has been under investigation for over 20 years. Before investigations into the LMSC, scientists and engineers were exploring the Land Mobile Radio Channel (LMRC). The first of these papers appeared in the 1950s when W.R. Young collected test data using a mobile radio system in New York City. Young may have been the first to suggest a Rayleigh distribution for the amplitude variations [12]. In the early 1960s, J. F. Ossanna took this topic farther by collecting some additional data and attempting to explain the results using reflected, interfering electromagnetic waves with random properties (reflected by randomly oriented flat surfaces with a uniform distribution). In his paper, Ossanna shows that his theory is generally good but does not match the data at all frequencies [13]. In 1968, Clarke combined the efforts of Ossanna, Young and others to propose a cohesive explanation of the Rayleigh scattering seen with a mobile radio channel [12]. While Clarke did not make a significant point of comparing urban and rural areas, most of the experimental data was collected in New York City. He expanded on Ossanna's idea that

"incident and reflected waves form an interference pattern through which the receiver moves" by including the effects of electromagnetic scattering [12]. While he did not specifically discuss shadowing or mention a log normal density, Clarke did suggest that the signal amplitude was a non-stationary Rayleigh process with a random mean. This effort was followed by many others. Investigators continued to collect test data at various frequencies and in varying environments [14, 15]. In the late 1970s, Suzuki proposed a stochastic model that was Rayleigh distributed but had a random log normal mean [16]. The Rayleigh probability density function (pdf) accurately modeled the signal scattering caused by buildings and the log normal mean accurately models the impact of shadowing (losing LOS to an obstacle). At nearly the same time, Hansen and Meno were pursuing similar thoughts regarding a Rayleigh density conditioned on a log normal input [17]. Like Suzuki, they suggested that the Rayleigh pdf accurately models the short term (high frequency) variation caused by multi-path and the log normal pdf captured the longer term shadowing. From [17], the pdf of the random variable that models the instantaneous signal power x is the Rayleigh pdf:

$$f(x) = \frac{1}{x_0} e^{\frac{-x}{x_0}}. \tag{1.1}$$

The log normal pdf is used to model the random variable that represents the mean instantaneous power x_0:

$$f(x_0) = \frac{M}{\sqrt{2\pi}\sigma x_0} e^{(\log x_0 - \log \tilde{x}_0)^2 / 2\sigma^2}. \tag{1.2}$$

Next the two probability density functions are combined to obtain the probability that the instantaneous power is below the receiver's threshold:

$$P(x \le x_T) = \frac{x_T}{\overline{x}_0} e^{(\sigma/4.3)^2}, \tag{1.3}$$

where x is the instantaneous signal power (Rayleigh),
x_0 is the mean of x (log-normal),
x_T is the receiver's threshold for detecting the signal,
$\overline{x}_0$ is the mean of x_0,

σ is the standard deviation of x_0, and
M = constant = 0.43.

Hansen and Meno went on to advance the discussion by deriving an equation to predict error rates.

In 1985, Loo applied many of the LMRC concepts to a "Land Mobile Satellite Link" [18]. Even though Loo's investigation was focused on performance in rural areas, his results have been applied in other environments. He proposed that foliage not only attenuates the LOS signal but also scatters the signal energy causing a Rayleigh distribution in signal amplitude. Like Suzuki, Loo found that LOS attenuation followed a log normal distribution. In general, the resulting density function is a complicated mixture pdf that requires numerical integration:

$$f(r) = \frac{r}{b_0\sqrt{2\pi d_0}} \int_0^\infty \frac{1}{z} \exp\left[\frac{-\left(\ln z - \mu\right)^2}{2d_0 - \dfrac{r^2 + z^2}{2b_0}} \right] I_0\left(\frac{rz}{b_0}\right) dz. \tag{1.4}$$

Loo found that shadowing dominates for large values of r, and the probability density function can be approximated as log normal. Similarly, scattering dominates when r is small, and the probability density function can be approximated as Rayleigh. As a result,

$$f(r) \approx \begin{cases} \dfrac{1}{r\sqrt{2\pi d_0}} \exp\left[\dfrac{-\left(\ln r - \mu\right)^2}{2d_0} \right] & \text{for } r \gg \sqrt{b_0} \\ \dfrac{r}{b_0}\left[\dfrac{-r^2}{2b_0} \right] & \text{for } r \ll \sqrt{b_0} \end{cases}, \tag{1.5}$$

where r is the signal magnitude at the receiver,
z is the log normally distributed amplitude of the LOS signal,
b_0 represents the average scattered power due to multipath,
μ is the mean of the log normal portion of the signal, and
d_0 is the variance of the log normal portion of the signal.

In 1991, Lutz et al., continued the investigation by proposing a mixture model [19]. Lutz proposed a summation of two distinct pdfs that are combined using a weighting factor A which represents the fraction of time that the LOS signal is blocked by an obstruction. When the LOS signal is not blocked, the signal envelope follows a Ricean distribution. When the LOS signal is blocked, the signal follows a Rayleigh distribution conditioned on the log normal variable S_0. This can be expressed by:

$$f_{lutz}(S) = (1-A)\cdot f_{Ricean}(S) + A\cdot \int_0^\infty f_{Rayleigh}(S|S_0) f_{\ln}(S_0)\, dS_0 \tag{1.6}$$

where S is the instantaneous signal power at the receiver,
S_0 is the short-term mean signal power at the receiver, and
A is the fraction of time that the LOS signal is blocked.

Only a few years later, Corazzo and Vatalaro suggested small improvements to Loo's and Lutz's models. While Loo assumed that scattered signal energy was essentially constant, Corazzo and Vatalaro suggested that, like the LOS signal, scattering was also impacted by the log normal shadowing [20]. They modified the scattering to be conditioned on the log normal shadowing. They applied similar logic to Lutz's model and conditioned the Ricean and Rayleigh pdfs on log normal means.

The investigation into pdf-based mixture models has continued into the 21st century. Several authors continue to investigate improvements to the basic concepts proposed by Lutz and others. Abdi et al., focused on simplifications to ease computing without sacrificing accuracy [21, 22]. Schodorf proposed a combination of three pdfs to better match the urban test data [5]:

$$f_{urban\_motm}(S) = (1-B_1-B_2) f_{hl}(S) + B_1 f_{\ln}(S) + B_2\left(\eta_1 S^{-\eta_2}\right), \tag{1.7}$$

where S is the received signal power,
f_{hl} characterizes the channel behavior with LOS signal,
f_{ln} characterizes the channel behavior without LOS signal,
B_1, B_2 represent the weighting of each pdf in the mixture model, and
η_1, η_2 are constants defining the power law for the third term.

The proposals of Lutz, Corazzo and Vatalaro, and Schodorf were all based on the simple linear combinations of individual pdfs. The end results of these combinations were new bimodal (or multi-modal) pdfs that could be used to understand the long term statistics of a LMSC: level crossing rates, average fade duration, etc. These were not necessarily the best way to understand the short term activity in the channel because they treated the non-stationary channel as stationary and did not take into consideration channel memory.

To address these short-comings and improve LMSC modeling even more, some scientists began investigating the use of Markov models. These authors still supported the basic concept of using different pdfs to model different environments (urban, suburban or rural, open), but instead of combining different pdfs into a single amalgam, they assigned a pdf to each state and transitioned between states using a hidden Markov model (HMM) [23, 24]. This idea is illustrated by Figure 3, where the state transition diagram is shown for a three-state Markov model. State S_1 is defined as an open environment without shadowing. States S_2 and S_3 are similarly defined as suburban and urban environments with incrementally more shadowing. The variables P_{ij} represent transition probabilities from state $\mathbf{S_i}$ to state $\mathbf{S_j}$.

As a mobile receiver moves, it tends to stay within a specific environment for a relatively long period of time. While the receiver is in state S_1, the LOS signal is not blocked and the channel can be modeled using a Ricean pdf. While the receiver is in state S_2, the LOS signal is occasionally attenuated and scattered by trees or small buildings. So the best pdf for this situation may be Rayleigh with a log normal mean, as suggested by Loo [18]. An appropriate pdf is also assigned to the urban state S_3. One possibility is Schodorf's suggestion that the urban environment is accurately represented by a mixture of three pdfs [5]. As shown by the transition diagram, the model is able to transition from any state to any state, showing that the vehicle can move from an open environment to an urban environment without passing through the rural environment.

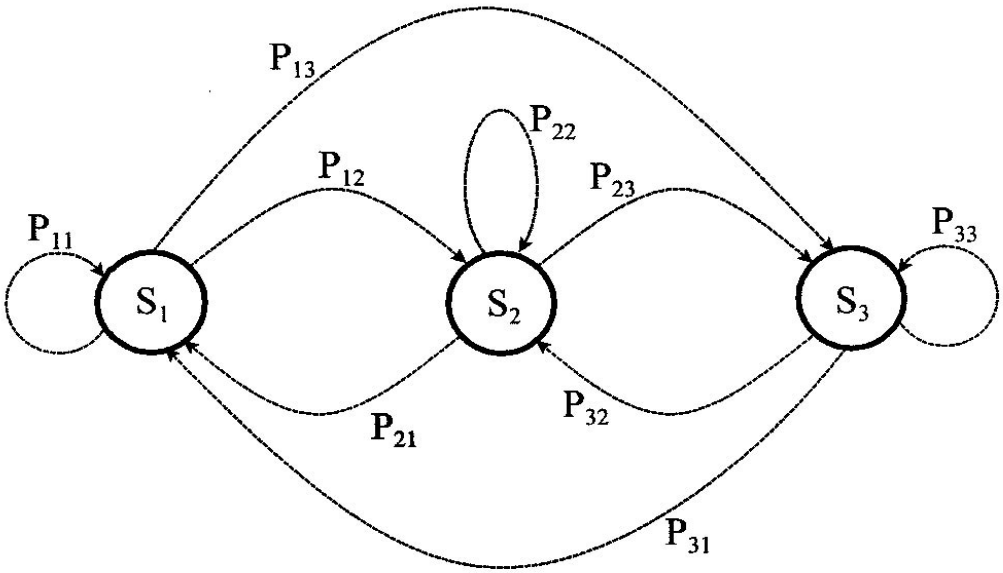

Figure 3. Markov model for LMSC proposed by F. P. Fontan et al. [24].

Figure 4 shows this model from a temporal perspective. The figure delineates between three orders of variation in the signal strength. The "fast" variations are usually attributed to multi-path fading (constructive and destructive interference of time-delayed signals). The "slow" variations are usually associated with specific shadowing events by foliage, terrain or buildings. The "very slow" variations are associated with changes in the environment.

The research presented in this dissertation narrows the focus to further investigate the urban state (S_3 above) because the urban environment is the most problematic for the COTM link being implemented in the TCA. As mentioned above, hidden Markov modeling is an important modeling approach for the general LMSC with states $S_1 - S_3$ representing the rural, suburban and urban environments, respectively. As part of this research, we have also applied another HMM internal to the urban state. An introduction to Markov models is provided in the next section and HMM modeling of the urban fade problem is discussed in Chapter II.

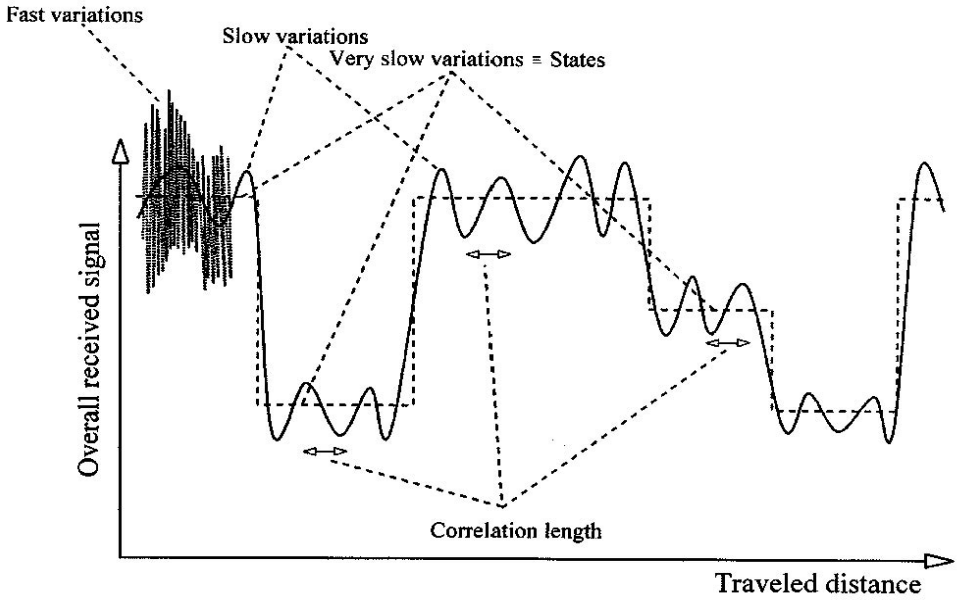

Figure 4. Short, medium and long-term variation of signal strength with time. Long-term ("very slow") variation is modeled by state transitions shown in Figure 3.

C. INTRODUCTION TO HIDDEN MARKOV MODELS

1. Basic Theory

"In order to be able to analyze and predict a system's performance, it is necessary to have some simplified theoretical description of the system's operation – *the system model*" [25]. Hidden Markov models are one possible choice of system model. Over the past 40 years, they have grown in popularity in many different fields of study besides communications engineering. Examples include ecology [26], computer network traffic [27], meteorology [28] and financial market analysis [29]. This growth in popularity is attributed to their rich "mathematical structure" that allows them to be applied to a wide range of applications and their ability to work well in practical simulations [31].

At the most basic level, there are two classes of Markov processes: those with a continuous state space and those with a discrete state space. This introduction will focus

on the latter group of Markovian processes with discrete states. The basic definition of a discrete state Markov model begins with a set of states, **S** = {S_1, S_2 … S_N}, where N is the number of states. The simplest version of any consequence is N = 2. With each time step, the model may change states or remain in the same state. The probability of each possible change is called a transition probability. A simple example is shown in Figure 5.

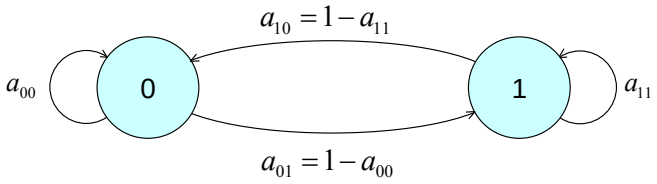

Figure 5. State transition diagram for two-state Markov model

Figure 5 is a pictorial representation of a two-state Markov model. The states are represented by the two circles with boldface **0** and **1** inside. The arrows that connect the two circles represent the possible transitions with each time step. Each arrow has an associated transition probability, labeled a_{ij}. The subscript *i* denotes the previous state and *j* denotes the next state. The arrow that leaves and returns to state **0** shows that the model can stay in state **0** as time progresses. The probability of the model doing this is represented by the variable a_{00}. The probability that the model transitions from state **0** to state **1** is a_{01}. Similar transition probabilities are shown for state **1**. It is important to note that the sum of the probabilities leaving any state is 1:

$$\sum_j a_{ij} = 1. \tag{1.8}$$

This relationship shows that a_{00} and a_{01} are related by $a_{01} = 1 - a_{00}$.

For a basic Markov model, the output is the state, $\mathbf{S_i}$, which can be directly observed. Rainfall can provide a simple example. We can define state $\mathbf{S_1}$ as "not raining" and state $\mathbf{S_2}$ as "raining". We can also define the time step as one hour. If there

is any rainfall during any fraction of an hour, then the model is defined to be in state $\mathbf{S_2}$ during that hour. By collecting weather information, we can determine the transition probabilities between "raining" and "not raining". The output of this model is the state itself. Once the model is completely defined, we can use it to analyze rainfall patterns including average rain duration, probability of six straight hours of rainfall, etc.

In contrast, the output of a ***hidden*** Markov model is the result of an additional random process whose statistical parameters are determined by the state. Instead of observing the state directly, the investigator sees the results of the second random process. Gilbert's theories regarding burst errors on a telephone line provide a simple example in the next section. Rabiner also provides a simple example regarding colored balls and urns [30, 31]. A simplified version can be based on two urns, each with three balls. Urn 1 has two red balls and one yellow ball. Urn 2 has two yellow balls and one red ball. The urns each represent a state. The colored balls represent the output of the model. When the model is in state **1**, a ball is drawn from Urn 1 and then put back. The only information that you, the observer, receives is the color of the ball. If the output is a red ball, you can not be certain which urn it came from. The state is "hidden". Probabilities suggest that the ball probably came from Urn 1. A reasonable output may look something like that shown in Table 1.

Table 1. Urn and colored ball example illustrates a simple discrete-time HMM.

Time (t)	1	2	3	4	5
Hidden State (Urn)	S_1	S_2	S_2	S_2	S_1
Output (Ball Color)	Y	Y	Y	R	R

This extra step of the HMM process is often called the "output", the "observation" or the "transmission". Likewise, the stochastic process that describes the observation is often called the transmission probability function or output process. This additional intra-state stochastic process inherent to hidden Markov models can also be either

discrete or continuous. The term Markov "chain" is used inconsistently to either describe all discrete state Markov models or is sometimes limited to those that have a discrete state space and a discrete output process. We will investigate both types of discrete state Markov processes and will use the term Markov chain for either.

While transition probabilities tell us how the model behaves with time, they do not provide the initial model state at start time $\mathbf{t_0}$. In addition to transition probabilities, it is important to know the initial state probabilities. These are usually denoted as a vector: $\mathbf{a} = [a_1, a_2, \ldots, a_S]$. It is easy to observe that $\sum_{i=1}^{S} a_i = 1$.

If a Markov model is homogenous, then its transition probabilities a_{ij} do not change with time. Similarly, if a Markov model is stationary, then its long term or stationary transition probabilities are equal to the initial state probabilities. Due to receiver mobility, the urban LMSC problem is guaranteed to be neither stationary nor homogeneous. As a vehicle moves from an open environment to an urban environment, the transition probabilities will shift dramatically. For this focused investigation into the network performance in an urban environment, we assume homogenous, stationary processes for the short term, where short term is defined to be on the order of a minute.

2. History

The first use of "Markov" models in communications appears to be by Shannon in 1948 [25]. At the time, Shannon called them "Finite State Channels". In other fields of study, these same models are know by other names like stochastic sequential machines or probabilistic automata [25]. In 1960, Gilbert used Markov models to model the performance of a real-world, switched telephone line [32]. He realized that the traditional assumptions of symbol independence and random error distribution were not very accurate for a switched telephone line. Instead, the errors tended to occur in bursts when a switch was thrown. In an effort to more accurately deal with this burst noise, he tried using a simple two-state Markov model to allow for channel memory. If an error occurred, it was statistically more likely to be followed by another error (a burst). If a bit

was received correctly, it was more likely that the next bit would be received correctly too. By using a model that reflected this memory, Gilbert was able to show increased capacity relative to a traditional binary symmetric channel. Gilbert went on to suggest some simple methods for estimating a channel's parameters: transition probabilities and state output probabilities. He also pointed out that a huge data-set was required for reasonable results. At this point, the concept of Markov modeling gained an audience within the communications community. It is important to note that while Gilbert did not use the term "hidden" to describe his Markov model, it was in fact a hidden Markov model. The output of the model was a binary error sequence (0s denoting no error, 1s denoting error). When an error was received, the model was clearly in state 2 ("Bad" state). When a correct bit was received, the model could be in either state. It was not possible to exactly match the error sequence with the state sequence, and the state sequence was therefore "hidden" by the output. We will revisit the Gilbert model in Chapter II when discussing Schodorf's attempts to apply the Gilbert model to the COTM empirical data.

A few years later, Elliott took Gilbert's proposal one step further and suggested a method to solve for probability of bit error [33]. Today, the terms "Gilbert" or "Gilbert-Elliott" are used by some authors to describe a burst noise channel [34, 35]. Several authors have built on Gilbert and Elliott's initial efforts and refined the analyses for advanced modulation or coding schemes [34, 35, 36].

An important milestone in the general development of HMM theory occurred in the 1960s when Baum and others made big strides in solving key problems. In 1966, Baum et al., proposed a solution to the most basic problem of estimating the model's parameters from a reasonably sized data set [37]. Again in 1970, Baum et al., solved another lingering problem: how to determine which state sequence was most likely considering the received information [38]. Baum's method is a specific version of the general Expectation Maximization algorithm, commonly called the Baum-Welch Algorithm (BWA). We will revisit this topic in Chapter II when discussing the parameter estimation of the COTM empirical data.

After these breakthroughs, scientists and engineers started to look at HMMs as possible solutions to their own problems in modeling complex real-world processes. One early area of application was speech recognition. Rabiner and others applied HMM theory to speech recognition and achieved some important successes [31]. Speech recognition is still an active area of continued research into HMMs. Even more recently, HMMs have been applied to modeling ecological patterns, neural networks, computer network traffic and operations research. During this period of growth, the use of HMMs in communications became more wide-spread and more diverse.

One of the primary benefits of Markov modeling communication channels is that they provide a more accurate representation of real-world channel memory where neighboring time samples are not truly independent. A second benefit is that Markov models are better at producing short-term, time-sampled, outputs. Some authors began using Markov models for the terrestrial wireless fading channel, where the fading was due to Doppler, weather, scintillation or multi-path. In 1995, Hwang and Moayeri suggested defining an N-state Markov model where each state represented a range of instantaneous signal-to-noise ratios (SNRs) [39]. As the instantaneous SNR varies with time, the Markov model varies between states S_1 to S_N. Table 2 shows a simple example for S = 6.

Table 2. Simple example of state definitions for HMM fading channel.

State	s_1	s_2	s_3	s_4	s_5	s_6
High SNR (dB)	−20	−10	0	+10	+20	$+\infty$
Low SNR (dB)	$-\infty$	−20	−10	0	+10	+20

Zhang and Kassam took this concept further by investigating packet-based systems and proposing a non-uniform partitioning of the SNR range [40]. They instead suggested a partitioning based on the average time duration in each state.

The basic idea behind using a HMM to model the LMSC is not new. In 1991, Wakana proposed a five-state Markov model for just this purpose (see Figure 6) [41, 42]. Two of the states were set aside to represent short and long-duration LOS shadowing by narrow and wide obstacles, respectively. The remaining three states defined performance when the LOS signal was not blocked.

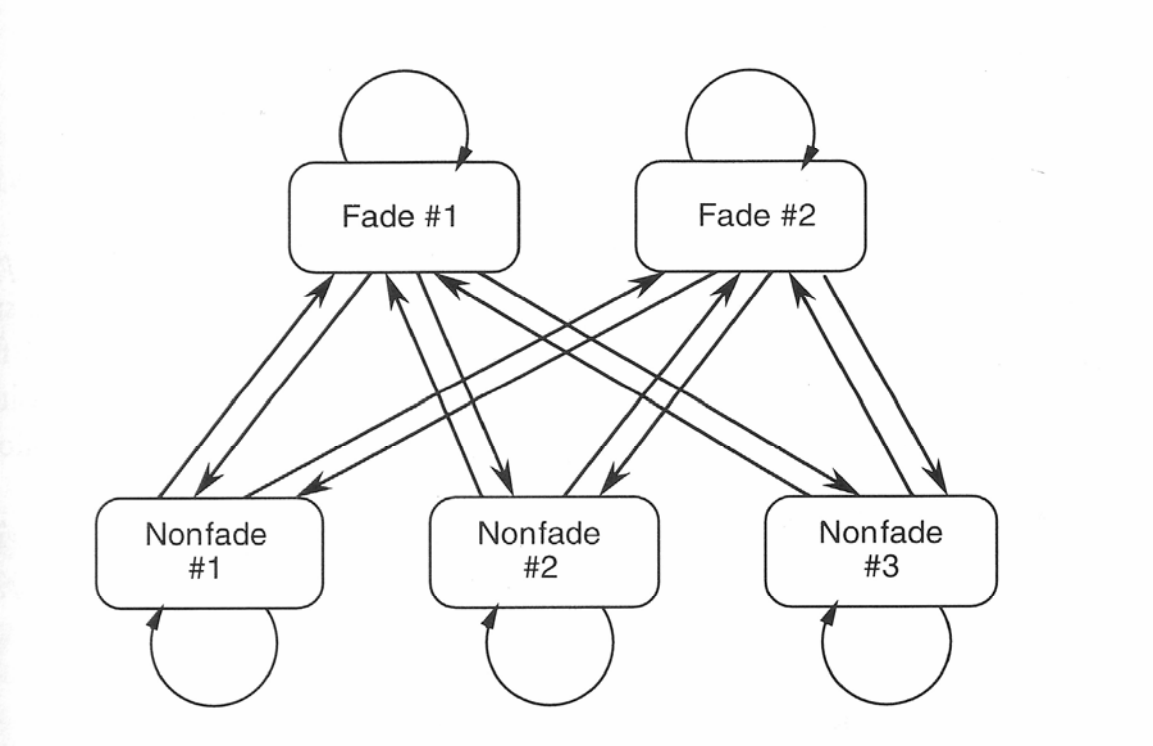

Figure 6. Five-state Markov model proposed by Wakana [41].

All five states allow for multi-path signal energy. Wakana imposed two transition restrictions: 1) non-fade states were not allowed to transition to other non-fade states and 2) fade states were not allowed to transition to other fade states. He then used a parameter estimation algorithm to solve for the transition probabilities. He defined eight variables which were related to state behavior and used these to define the following state transition probabilities:

p_1 = Probability of being in Fade State 1,
q_1 = Probability of transitioning from Fade State 1 to Fade state 1,
q_2 = Probability of transitioning from Fade State 2 to Fade state 2,

p_2 = Probability of being in Non-fade State 1,
p_3 = Probability of being in Non-fade State 2,
q_3 = Probability of transitioning from Non-fade State 1 to Non-fade state 1,
q_4 = Probability of transitioning from Non-fade State 2 to Non-fade state 2, and
q_5 = Probability of transitioning from Non-fade State 3 to Non-fade state 3.

Babich and Lombardi extended Wakana's concept to define the equations and relationships for a general LMSC with N states [43]. Later, Fontan et al., proposed the three-state HMM discussed in Section B. The research presented in this dissertation assumes that the general LMSC channel is being modeled using an approach similar to that proposed by Wakana [41] or Fontan et al. [24] where the different environments are represented as state changes in a macroscopic HMM. The research then focuses on the state that represents the urban environment and inserts an additional continuous transmission hidden Markov process into the urban state.

D. DISSERTATION OVERVIEW: MODELING THE URBAN LMSC

The research of this dissertation is focused on mitigating the urban LMSC/COTM signal fade problem and its associated impacts on network performance. A central task towards achieving this goal is the development of a time-based simulation of the COTM channel. This model is expected to be valuable is several different ways. First, the model can provide an inexpensive method for producing additional data, perhaps as an input to a filter or algorithm. It is expensive to drive mobile test platforms around urban environments to collect additional LMS data. Second, a model can provide additional insight into the physical situation and suggest possible solutions that can be difficult to see when examining empirical data. More specifically, a model is often the basis for developing a prediction approach, which is one of the follow-on tasks here. Finally, the model can provide an inexpensive means to investigate the benefits achieved with a possible solution.

Two distinct modeling approaches have already been suggested in relevant literature and summarized in Sections B and C. The first is a probability density model and the second is a Markov model with an emphasis on the HMM variation. We consider

the second approach to be more appropriate for the Urban COTM problem. This dissertation focuses on improvements to Markov modeling of the Urban Land Mobile Satellite channel (ULMSC) and using that model to improve network performance. The remainder of Section D is devoted to introducing the three areas of study: 1) using empirical data to build a model, 2) using the model to build a network simulation and 3) using the data and model to investigate fade prediction and associated network improvements.

1. Empirical data

As discussed in Section I.B, the Land Mobile Satellite Channel has been studied by many authors and several different models have been proposed [18, 21, 24]. For most of these investigations, the mobile-user has a non-directional antenna that collects a significant amount of multi-path signal power even when the LOS signal is blocked by an obstruction. As mentioned in Section I.A, the COTM user will have a highly directional antenna that essentially rejects multi-path signal power. For this situation, the LOS signal is considered the only input to the receiver. Soft obstructions, like trees, have been shown to attenuate the LOS signal (−15 dB), but the receive antenna is still able to detect the LOS signal [5]. Hard obstacles, like buildings, usually block the entire signal (−30 dB), and the link is temporarily broken [5].

Schodorf attempted to model the urban COTM LMSC link with a simple two-state Markov or Gilbert model with limited success, and key results are shown in Figure 7 [5]. Schodorf investigated three different approaches to modeling the COTM channel as a Markov chain. The first and simplest approach is a simple two-state Markov model where one state is “LOS blocked” and the other state is “LOS not blocked”. This approach is not a HMM because the model state is observable from the model output. Results showed that this approach does not produce an acceptable match to the connection and fade durations found in the collected data. The other two approaches use a two-state Gilbert model which is a HMM with a Bernoulli transmission function (see Section I.C.). The two states are known as “Good” and “Bad”. The “Good” state is

defined to have a clear LOS signal (no shadowing). The "Bad" state is defined to mean that the LOS signal is blocked to some degree, where the amount of shadowing is a random process. Sometimes the shadowing is enough to reduce the signal power below the receiver's fade threshold, and sometimes it is not. Schodorf used this model twice. The first time only connection durations were used to determine the model's parameters. The second time he used both connection and fade durations. Plots resulting from this final approach are labeled "joint opt." in Figure 7.

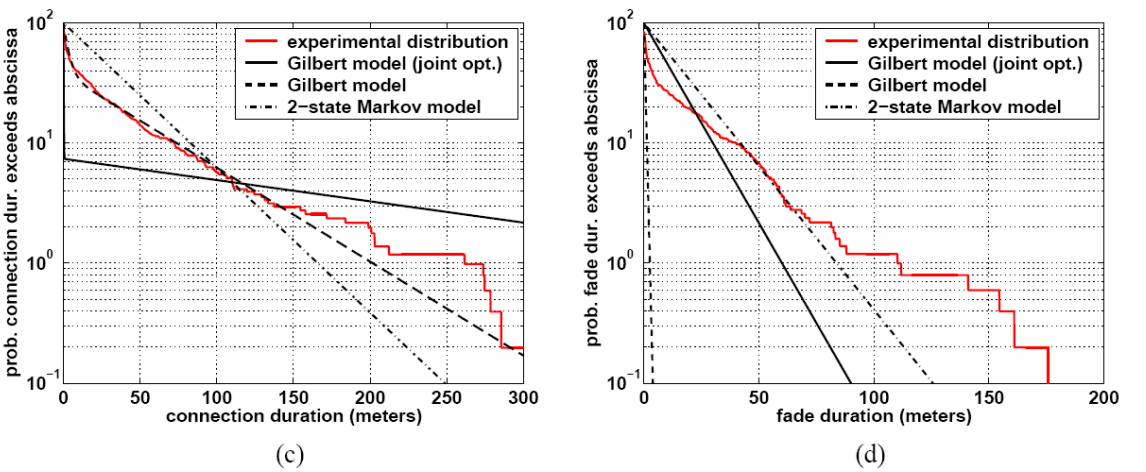

Figure 7. Complementary CDFs showing second order statistics from experiments and modeling attempts [5].

As discussed in Sections I.B. and I.C., other authors have proposed more complicated Markov models for non-directional systems. This dissertation starts with an effort to improve the accuracy of Schodorf's Markov model in matching real COTM data. The data used in this research was recently collected from an operational MILSTAR satellite by investigators at Lincoln Laboratory in Lexington, Massachusetts [5]. The data was collected using a mobile military vehicle equipped with a 12 in. directional antenna as planned for the TCS COTM link (see Figure 2). Data was collected in urban, rural, and open environments. The MILSTAR-on-the-Move (MOTM) test project and associated empirical data are discussed in more detail in Chapter II.

Several possible improvement paths have been mentioned in Sections B and C including an increase to the number of states, an increase in channel memory, and a more complex HMM with intra-state, weighted, mixture density functions. The improvement path explored for this project is based on the Baum-Welch parameter estimation algorithm. The Baum-Welch parameter estimation algorithm is mentioned by Schodorf, but avoided in favor of Gilbert's simplified parameter estimation approach. The results of the Baum-Welch parameter estimations are presented in Chapter II.

2. Network Simulations, Packet Routing Protocols and Performance Estimates

The primary investigation of this research is the development, validation and analysis of a simulation representing a mobile communications node situated in the urban environment and experiencing long, deep fades in signal strength. Two packet routing protocol modifications are considered to explore possible performance improvements. In addition to simulating the mobile-to-satellite COTM link, each model also simulates a local mobile-to-mobile ad-hoc network operating in parallel. The additional ad-hoc network is simulated because it provides a possible solution to the network outages caused by satellite blockage. After connecting the mobile ad-hoc network to the satellite network, the Neighbor protocol can be implemented to exploit the additional routing paths through the combined network. Chapter III describes each model in detail and Chapter IV presents simulation results, compares model performance and proposes future improvements. The simulations include embedded Markov models to represent the signal strength state transitions between "clear" and "fade" as introduced earlier. These Markov models are partially-based on the parameter estimation results discussed in Chapter II.

3. Urban LMSC Prediction and Control

Another fade mitigation path explored as part of this investigation involved predicting urban link fading and using the predictions to modify link activity for better overall performance. If successful, the investigation would identify an optimal recursive

time-varying control system similar to the Kalman, LMS or RLS algorithms frequently found in control and signal processing applications. A limited discussion of two attempts to implement an optimal recursive feedback algorithm as a solution to the long fades inherent in the urban LMSC is presented in Appendix A. The goal of the resulting algorithm is to improve channel throughput by predicting the COTM channel state (e.g., connection or fade) and modifying packet transmission appropriately. One response is to suspend transmission when a long duration blockage is predicted to minimize packet retransmission. Another response is to determine which packets are likely to be blocked and retransmit them. A prediction based solution would be most effective if implemented at the satellite end of the channel but would require a prediction capability beyond the path delay. Instead, prediction at the mobile node, which is easier due to proximity to the fade, is the focus of Appendix A. Because the linear prediction filters investigated as part of this research did not provide accurate and timely fade prediction, the algorithms were not added to the simulation presented in Chapters III and IV. The results and supporting discussion are provided in Appendix A for reference but are not considered a successful mitigation strategy for the urban COTM fade problem.

E. ORIGINAL CONTRIBUTION

Several original contributions to the communications field associated with urban Land Mobile Satellite channels with an emphasis towards a mobile directional antenna that causes a severe and extended fade of −30 to −40 dB below LOS are presented in this dissertation. The first contribution is the development of a Markovian COTM channel model that is significantly more accurate than previous attempts. This COTM channel model is used as a foundation for additional progress towards mitigating the COTM fade impacts on network performance. The second original contribution is the development and validation of a GIG network simulation that includes a local-area mobile ad-hoc network (MANET) and numerous independent COTM channels. This simulation provides a low-cost source of additional data, estimates of channel and network performance and a vehicle for investigating possible mitigation strategies. The final

contribution of this research is the determination that linear prediction algorithms require a high frequency signal input to have a chance at successfully predicting a channel fade.

F. CHAPTER I SUMMARY

In this chapter, we introduced and provided relevant background for the two central research goals of this dissertation: 1) developing an improved urban COTM channel model and 2) finding a fade mitigation strategy that improves network throughput and packet latency. In the next chapter, we address the first goal by analyzing the empirical test data and solving for an improved hidden Markov model.

II. EMPIRICAL DATA AND THE BAUM-WELCH ALGORITHM

The idea and history of Markov modeling the general LMSC was introduced in Chapter I. In this approach, each state of the Markov model represents a type of shadowing and attenuation environment (e.g., open field, rural, suburban, urban). The idea of using additional embedded Markov models within each of these Markov states was also introduced in Chapter I. Schodorf implemented this idea by proposing Gilbert models for the open, rural and urban environments [5]. In this modeling attempt, the urban environment is the most poorly represented by a Gilbert model, and additional improvements seem necessary. The focus of this chapter is on finding a more accurate model of the urban COTM channel than the Gilbert model proposed by Schodorf.

One important application of an accurate channel model is possible inclusion in the discrete-time, urban COTM simulation presented in Chapters III and IV. Of course, the empirical data could be used as a direct input to the simulation; but the data set is fairly limited in duration and only reflects a single driving path through a single city. The results of such a simulation would have limited applicability. This approach would be more appropriate if the test vehicle was repeatedly driven along different paths in multiple cities. In the end, such an approach would produce a large and unwieldy database of signal strength as a function of time. A better approach is to use the empirical data set to develop a stochastic Markov model that can be manipulated to represent a variation of paths through different cities. The Baum-Welch algorithm plays a key role in converting the empirical data into parameters of a Markov model.

The investigation starts with a discussion of the empirical data collected by Lincoln Laboratory in support of the TCS program. Section II.B highlights some supporting data processing steps necessary for input to the Baum-Welch algorithm. Section II.C provides an introduction to Baum-Welch parameter estimation, and the chapter finishes with an in-depth discussion of the various parameter estimation approaches attempted and their results.

A. EMPIRICAL DATA

As part of the overall effort to understand and model the COTM link of TCS, Lincoln Laboratory built a hardware test platform and used it to collect empirical data. ESC-TR-2003-051 Technical Report 1087 discusses this testing in significant detail [5]. A summary follows. The new TCS satellite has not yet been built so testing had to be accomplished with the best available surrogate; Schodorf selected MILSTAR to be that surrogate due to its many similarities with the planned TCS [5].

The Lincoln Lab test program included broader goals than just urban channel characterization. Their goals included channel characterization and modeling, antenna pointing performance, and packet protocol performance in open, rural, and urban environments. Data was collected in three different locations to investigate the three different environments; open and rural routes were located on Ft. Devens, MA, and the urban route was located in Boston, MA. Because the focus of this dissertation is limited to the urban communications channel, the discussion here is focused on the channel characterization (signal strength) subset of the urban data set collected in Boston. Additional information regarding the antenna pointing tests, packet protocol tests and open / rural test environments, may be found in the original technical report [5].

The urban channel characterization data was collected on four separate occasions: October 17, October 22, October 31, and November 8, 2002. Over these four dates, eight test runs were completed. To provide continuity, Schodorf's nomenclature is used in this work, and the eight tests are referred to with letters A-H. All test data sets include a measurement of the signal-to-noise ratio (Pr/No) taken every 0.1 seconds. Additionally, some data sets also include a measurement of the vehicle velocity taken every 1.3 seconds (on average). Table 3 shows the number of samples collected during each test. As an example, Figures 8 and 9 show an excerpt of the signal strength and velocity data collected as Data Set B. It is important to note that the signal strength data presented in Figure 8 has been shifted to be relative to LOS.

Table 3. Empirical data set from Lincoln Lab COTM testing.

Date	Test	Number of samples: Signal strength	Number of samples: Velocity	Velocity data available?
Oct 17	E	6220	0	N
Oct 22	F	6160	0	N
Oct 31	A	5600	4442	Y
Oct 31	B	4200	3520	Y
Oct 31	C	6920	5535	Y
Oct 31	D	6160	4350	Y
Oct 31	G	7200	5685	Y
Nov 8	H	5320	0	N

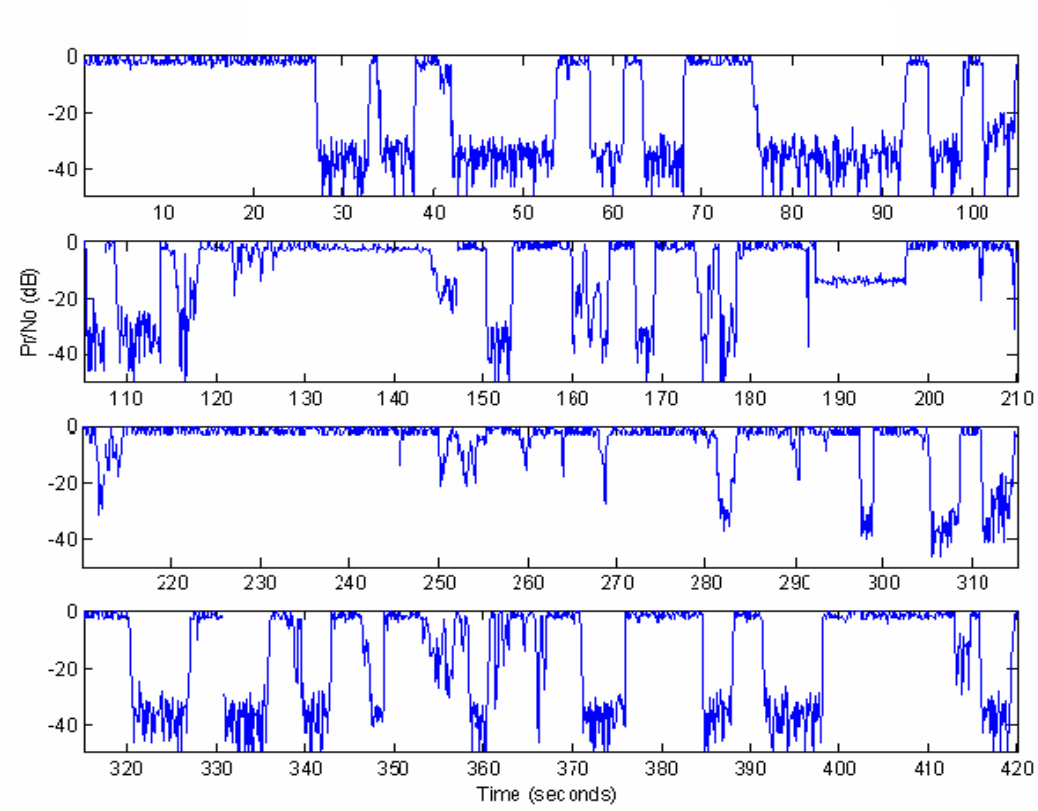

Figure 8. Signal strength vs. time from Data Set B (relative to LOS).

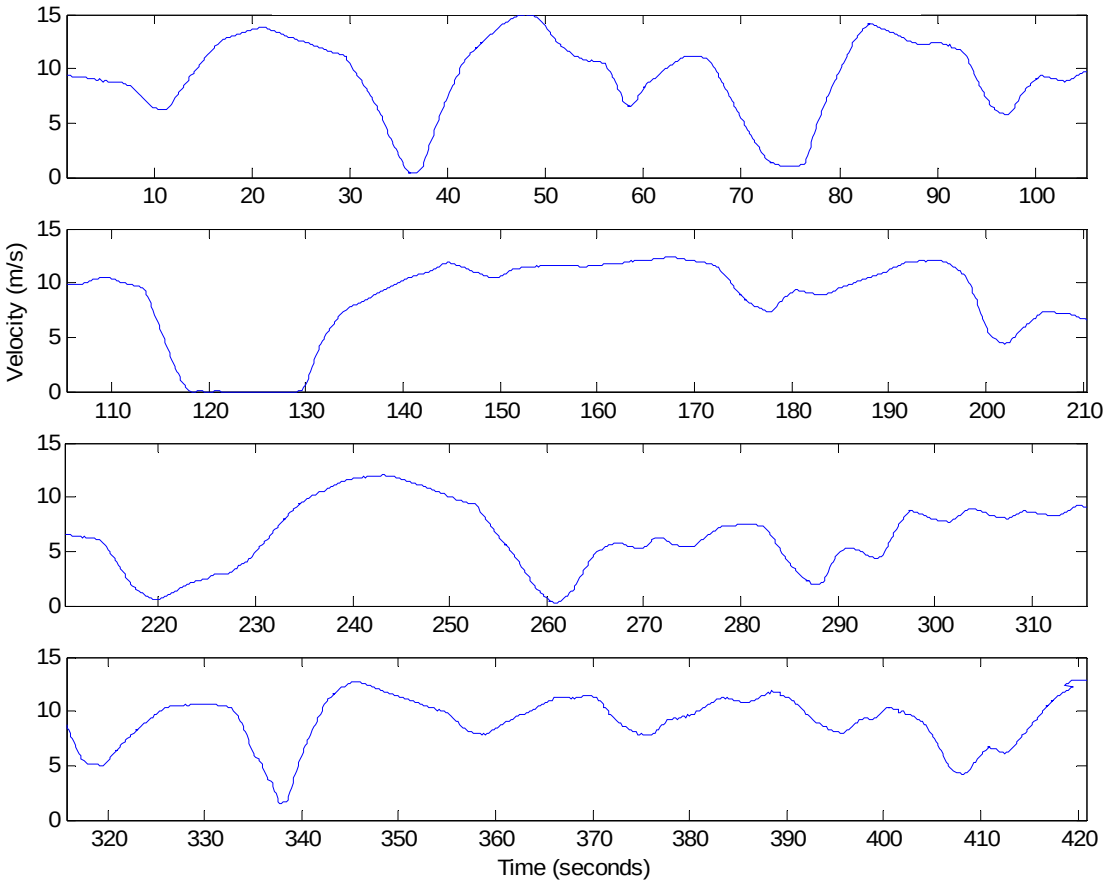

Figure 9. Vehicle velocity vs. time from Data Set B.

An important aspect of the test program, and its resulting data set, is that each test was made on the same path through the same city. This means that the correlation between data sets is high, and the additional data sets do not provide completely new information. An obvious way to improve the analysis and modeling of the urban COTM link is to broaden testing to multiple driving routes through multiple cities.

Another important aspect of the data is the fact that signal strength and velocity measurements were not taken simultaneously. This has an impact on data analyses which is discussed in Section 2.B.

1. Histogram

An important method for analyzing empirical data is the histogram, which can approximate the probability density function (pdf). Signal strength data from all eight data sets (A-H) was used to generate the histogram, shown in Figure 10 with a bin size equal to 0.5 dB. This specific bin size was the smallest bin size that did not produce a significant number of empty bins. Unlike the data presented in Figure 8, this signal strength data is presented in its original empirical form and is not shifted to be relative to LOS.

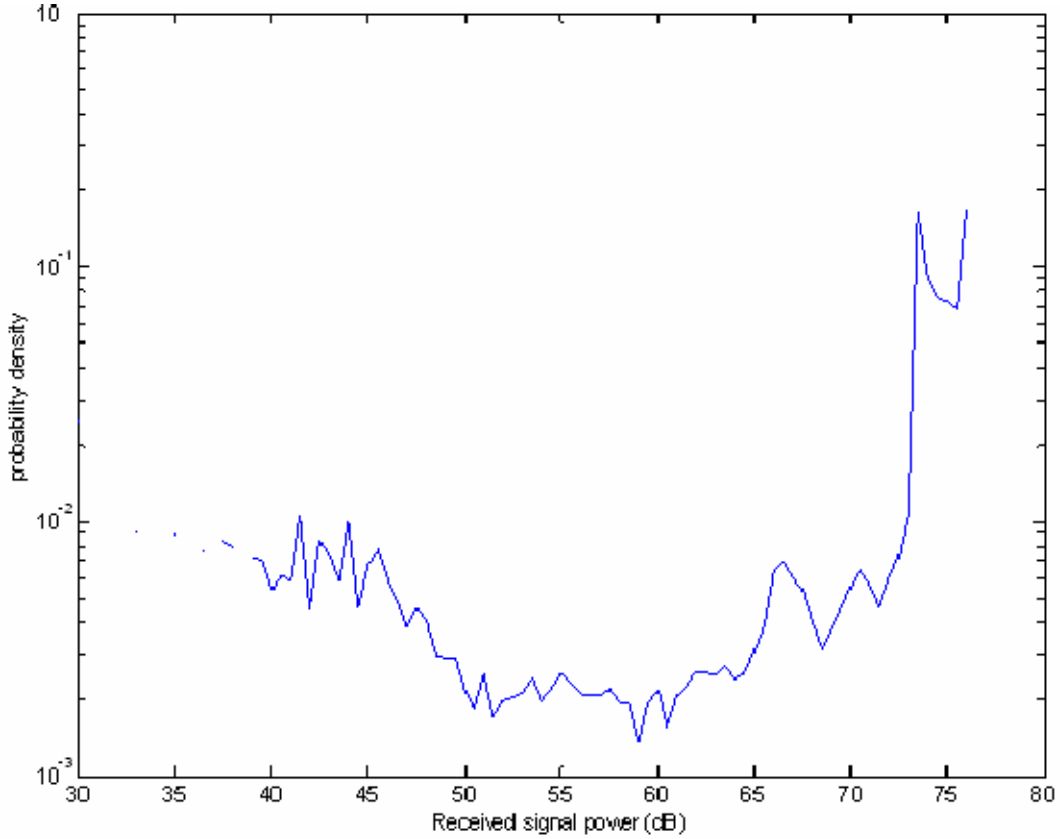

Figure 10. Histogram (raw data; no processing).

The largest histogram peak is centered near 75 dB and actually shows up as a double peak separated by about 3 dB. This double peak is expected and represents clear communication between the mobile vehicle and the satellite. The next largest feature occurs around 40-45 dB. This peak accounts for time steps where the receiver is

experiencing a deep fade. It is important to note that the region between these two peaks, which extends from approximately 50-65 dB, shows a low probability of receiving partial signal fade more associated with foliage. The number of peaks in the histogram (pdf) is one method for determining the appropriate number of Markov states. As Schodorf notes, the empirical data is influenced by the design of the hardware test bed (shown in Figure 2) and does not reflect theoretical expectations [5]. The front end of the mobile test bed includes a hard-limiter that modifies the true density function by saturating and truncating the highest signal powers. One side effect of this hard-limiting front end are the two neighboring peaks near 74 and 76 dB. At the other end of the histogram, the test bed is not able to detect signal power below its noise floor, which is approximately −40 dB. This prevents the histogram from tapering off smoothly below −40 dB.

2. Interarrival or Dwell Times

Another important parameter of the fade process is the time between state transitions. Some authors refer to this parameter as *dwell time* [5]; others refer to it as *interarrival time* (time between events) [27]. This dissertation will use either term interchangeably.

Using Kendall's notation for a simple M/M/1 queue[1], we model the interarrival time as an exponential random variable where the probability of an event after k time steps is

$$\Pr(n = k) = \frac{e^{-k}}{k}. \tag{2.1}$$

For a Markov model with two states, this relationship can be modified to reflect the actual transition probabilities of the problem. The probability of being in state i for k time steps is

$$\Pr(s_t \ldots s_{t+k} = i) = a_{ii}^k \left(1 - a_{ii}\right). \tag{2.2}$$

[1] In 1953, David G. Kendall introduced the A/B/C notation of queueing theory. In this notation, A is the interarrival time distribution, B is the service time distribution and C is the number of servers or queues. Common distributions are "M" for Markovian, "E" for Erlang, or "G" for General.

For this specific problem, we have two defined states: connection and fade. It is important to investigate the durations of these two states separately. Histograms of the fade and connection times are shown in Figures 11 and 12. In addition to the histograms, Figures 11 and 12 show the complementary CDF for each state duration. The complementary CDF format allows for better comparison with Schodorf's results presented in Figure 6. Summary statistics for the original data set are also shown in Table 4.

Table 4. Summary statistics for basic data set.

(all data in seconds)	Mean	Stand. Dev.	Maximum
Connection durations	5.39	11.05	99.1
Fade durations	1.68	2.78	30.9

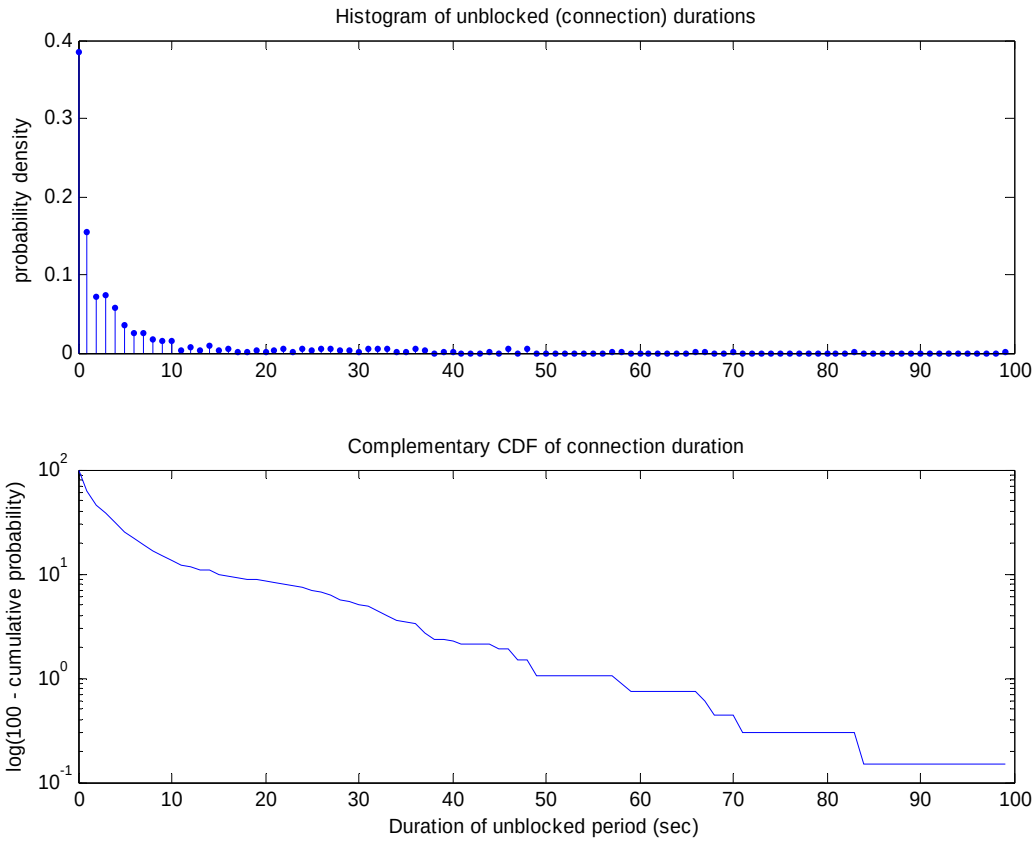

Figure 11. Histogram (top) and CDF (bottom) of connection duration (unsmoothed).

The complementary CDFs are shown on a logarithmic scale for easy comparison to exponential or geometric processes, which would appear linear. While the empirical dwell times aren't perfectly linear, they do appear to be roughly approximated as linear. This is another indication that the urban COTM channel fade process can be approximated as Markovian.

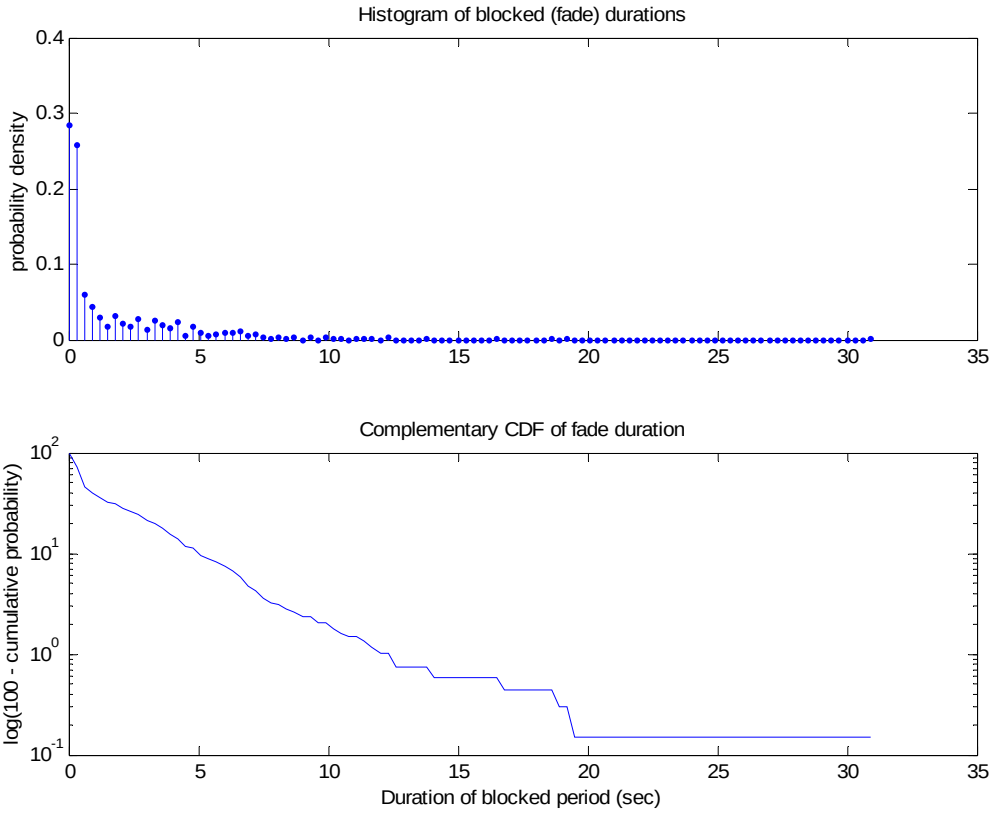

Figure 12. Histogram (top) and CDF (bottom) of fade duration (unsmoothed).

It is important to note that there are a large number of short duration transitions to State 0 (fade). We propose that these do not represent buildings blocking the satellite's signal. Assuming that the event is caused by a physical object between the mobile receiver and the satellite, we can find that a 0.1 second duration matches an object that is on the order of one meter in width. This approximation assumes that the mobile transmitter is moving at a speed of 10 m/s (~22 mph), which is a reasonable assumption

based on the velocity data presented in Figure 9. The next section discusses the application of a smoothing filter to minimize these extremely short transitions.

3. Smoothing

As mentioned above, the original data set contains a significant number of short duration state transitions, where short duration is defined as 1-2 time steps (0.1 – 0.2 seconds). No effort was made during testing to correlate these events with physical phenomena, but it is believed that these events are trees, light poles or other tall, thin objects commonly found in urban environments. This type of short duration event is also found in rural and suburban environments [5] and is not a challenging phenomena for mobile urban communications. The challenging phenomenon that is present only in urban environments is the tall building. Buildings are significantly wider and deeper than trees and utility poles. They cause deeper and longer fades of the signal and are nearly impossible to correct with interleaving or FEC. In an effort to focus the research on the buildings, we applied a median filter of length 3 to the data to remove the shortest (0.1 second) state transitions. The median filter reduces the number of single time step state transitions and focuses the analysis on longer duration effects. Table 5 is an update of Table 4 and shows the summary statistics for filtered data. Figures 13, 14 and 15 are updated versions of Figures 10, 11, and 12, respectively. The modified data set was then used for additional processing including parameter estimation.

Table 5. Summary statistics for smoothed data set (window = 3 samples).

(all data in seconds)	Mean	Stand. Dev.	Maximum
Connection durations	7.92	13.80	112.5
Fade durations	2.44	3.21	30.9

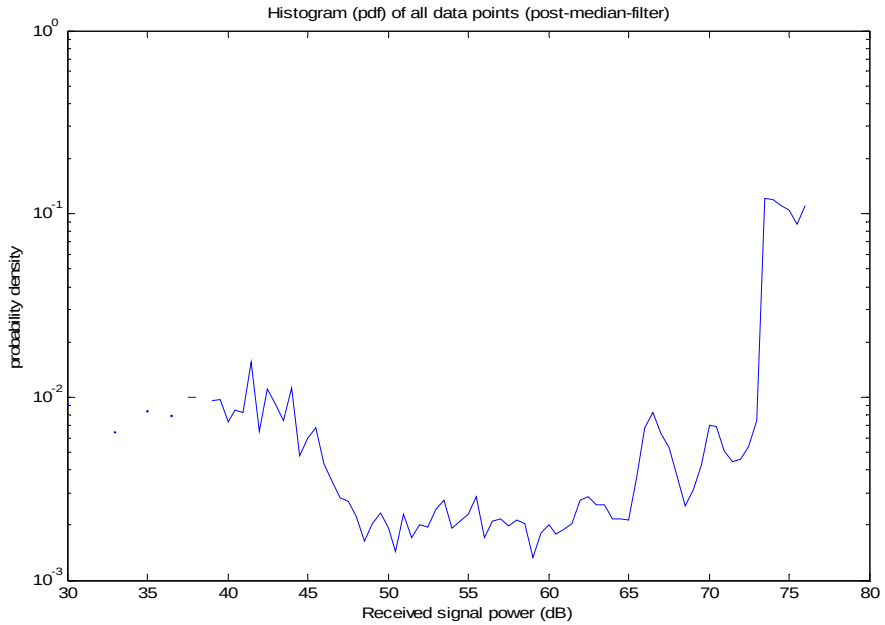

Figure 13. Histogram of received signal power after smoothing.

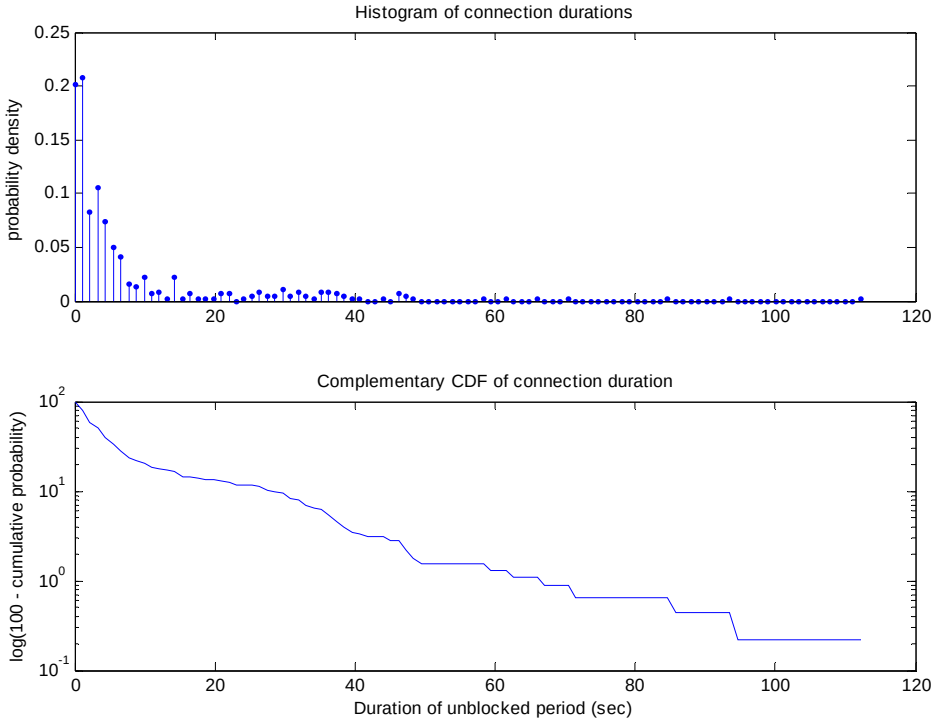

Figure 14. Histogram (top) and CDF (bottom) of connection duration (median-filtered).

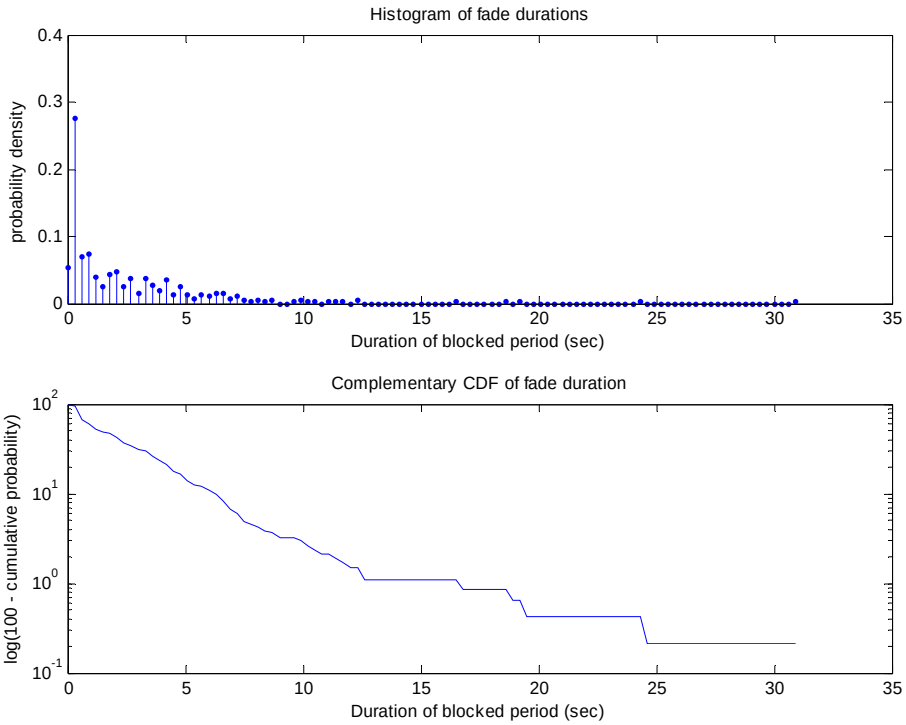

Figure 15. Histogram (top) and CDF (bottom) of fade duration (median filtered).

B. CONVERSION FROM TIME SAMPLING TO SPATIAL SAMPLING

As mentioned earlier, both the signal strength and velocity data were originally collected and presented as functions of time, *s(t)* and *v(t)*. The data analysis already presented was completed in the time domain. The parameter estimation process (next section) and the simulation presented in Chapter III are both based on time domain input vector for signal strength. Time domain analysis is clearly important to several facets of this investigation. But it is also important to analyze the data from the spatial perspective.

In the time domain, the signal vector represents a specific case of a mobile moving through an urban environment using a specific speed profile. Buildings are fixed spatially, and analyzing the problem from this perspective should provide insights regarding average building width and average gap width in meters. After converting the empirical data to the spatial domain, we can also generate additional signal strength

vectors in the time domain by applying a different mobile velocity pattern. The spatial perspective is also important for to the fade prediction investigation because the scattering of signal energy is a function of building-mobile geometry. This topic is discussed further in Appendix A.

Note that these two perspectives are related by the vehicle's velocity. An important aspect of the spatial perspective is that it provides results that are independent of vehicle velocity. In reality a gap between two buildings may be very small; let us use two meters for an example. This small gap provides a 0.2 second communication window when the vehicle is moving at ten meters per second. Unfortunately, 0.2 seconds is not enough time to send a packet and receive an acknowledgement. The utility of that small window will usually be fairly low. On the other hand, the vehicle may stop for a few seconds in that gap between buildings. In this situation, the connection window appears to be much longer, and several packets can be transferred with acknowledgements received. While they occur in the same physical situation, the results are significantly different and highly dependent on vehicle velocity.

It is important to note that some data collection days did not include vehicle velocity. The reasons for this are unclear, but without velocity data, we cannot convert the time samples to the spatial domain. For the remainder of this procedure, we will limit the analysis to Data Sets A, B, C, D and G.

The core of the conversion algorithm is a simple Riemann Sum. We can estimate the distance traveled during a sample period, By assuming that the velocity is constant over each sample period ($\Delta\text{distance} = \text{velocity} \times \Delta\text{time}$). By collecting these small distances in a running total, we are performing a simple integration using the Riemann Sum approach. The result of this simple algorithm is distance traveled versus time. An example is shown in Figure 16.

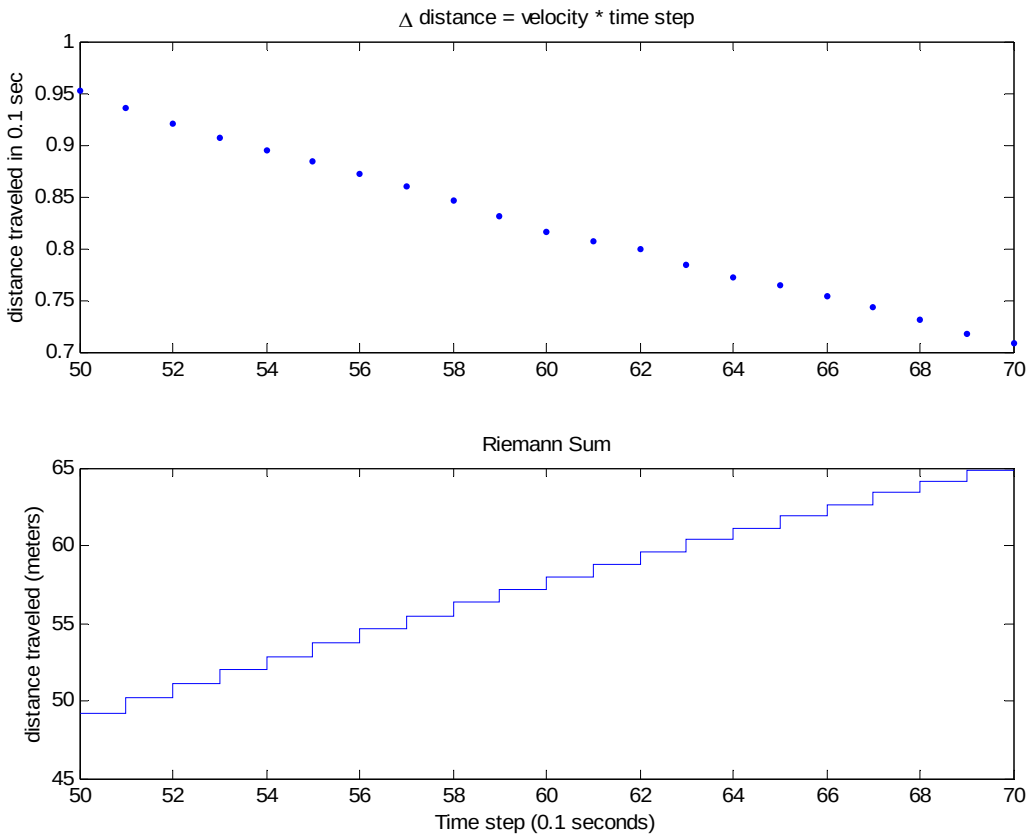

Figure 16. Riemann Sum of distance traveled.

The Riemann Sum algorithm provides vehicle location estimates at the same sample times that the velocity was collected. This causes a small discrepancy in the comparison of vehicle location and signal strength, because the velocity and signal strength were collected at different sample rates. This discrepancy can be corrected by resampling (or interpolating) one data vector to match the other. For this analysis, the velocity vector was resampled from 7.6 Hz to 10 Hz for direct comparison with signal strength.

After matching the signal strength and vehicle location in time, we can drop the relationship with time and use the direct comparison of signal strength as a function of distance traveled.

The impact of this data conversion are shown in Figures 17 through 19. Figure 17 shows the data from the temporal perspective. Figure 19 shows the same data from the spatial perspective. Figure 18 shows the vehicle velocity during this time period and is the key to converting between the time and distance dependence. Each figure has yellow and red ellipses which highlight two different connection periods. The yellow ellipses highlight a connection period of approximately 1.5 seconds which is enough time to send several packets and receive acknowledgements. During this period, the vehicle is moving at a rate of only ~2.5 meters per second. Figure 19 shows that this physical opening is only four meters across, but the connection time was significant because the vehicle was moving slowly. This phenomenon is even more apparent in the period highlighted by the red ellipses. In this case, the connection is approximately three seconds long in time but only one meter wide. Looking at Figure 18, we see that the cause of this long window is that the vehicle has essentially stopped. This situation puts several time samples directly on top of each other in the spatial perspective.

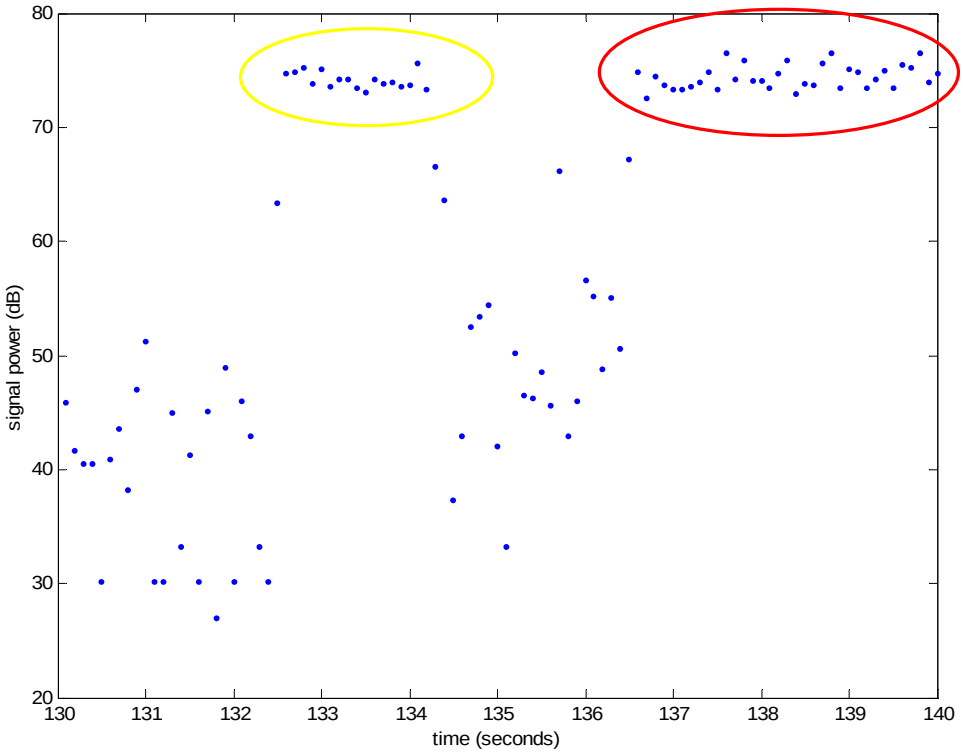

Figure 17. Data from temporal perspective (signal strength vs. time).

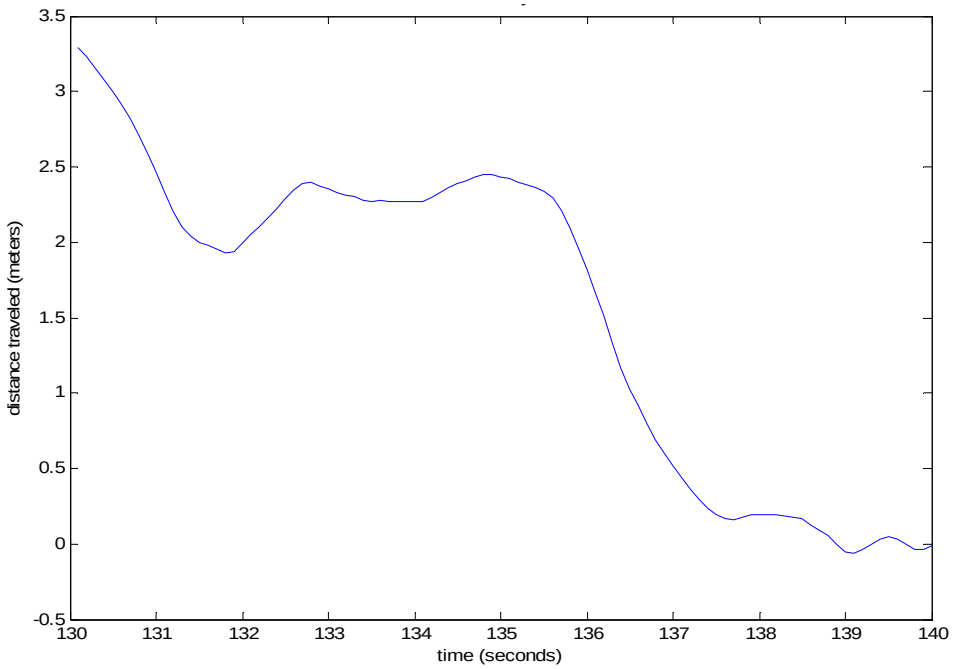

Figure 18. Velocity: v(t).

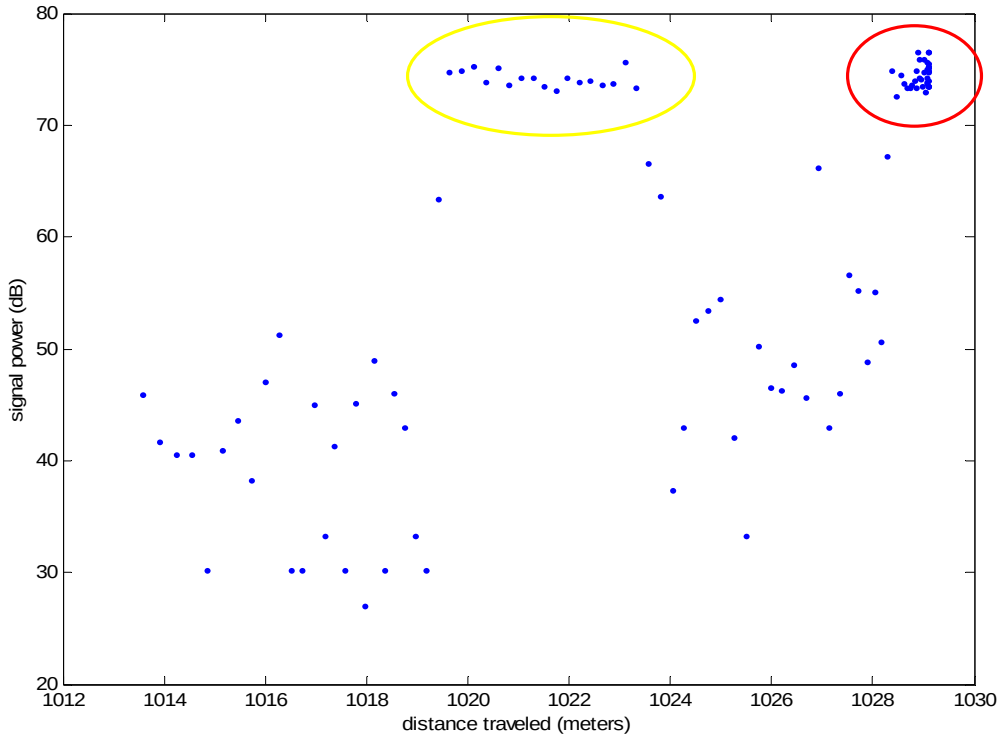

Figure 19. Data after conversion (signal strength vs. location).

The last step of data conversion is to interpolate the data to a constant separation distance step size of 0.25 meters. This is important for building a histogram (e.g., probability density function) and also important to developing a discrete-time Markov model. Data sets A, B, C, D and G were converted to spatial sampling using this process. The summary statistics are shown in Table 6 and Figures 20 through 22 for direct comparison with the temporal data.

Table 6 shows a mean fade duration of 16 meters (~52 feet), which seems to be a reasonable width for an average building. Of course, this average is not based only on buildings and is skewed smaller by the inclusion of very narrow scatter sources like light poles and street signs. It is also interesting to note a shift in the ratio of connection statistics to fade statistics. While the mean and maximum connection durations are 3.2 times longer than the mean and maximum fade durations in time, the equivalent ratio is only 2.2 in distance. This odd relationship implies that, on average, the vehicle velocity is lower during connections than fades. Perhaps this is an impact of city traffic signals, where the vehicle generally has clear access to the satellite while sitting still for a red light. In any case, this ratio shift highlights the importance of analyzing the fade process in both time and distance. Clearly, the velocity profile of the mobile has a substantial impact on the channel performance.

Table 6. Summary statistics for spatial data set.

(all data in meters)	Mean	Stand. Dev.	Maximum
Connection durations	36.4	62.3	379.50
Fade durations	16.0	26.2	188.25

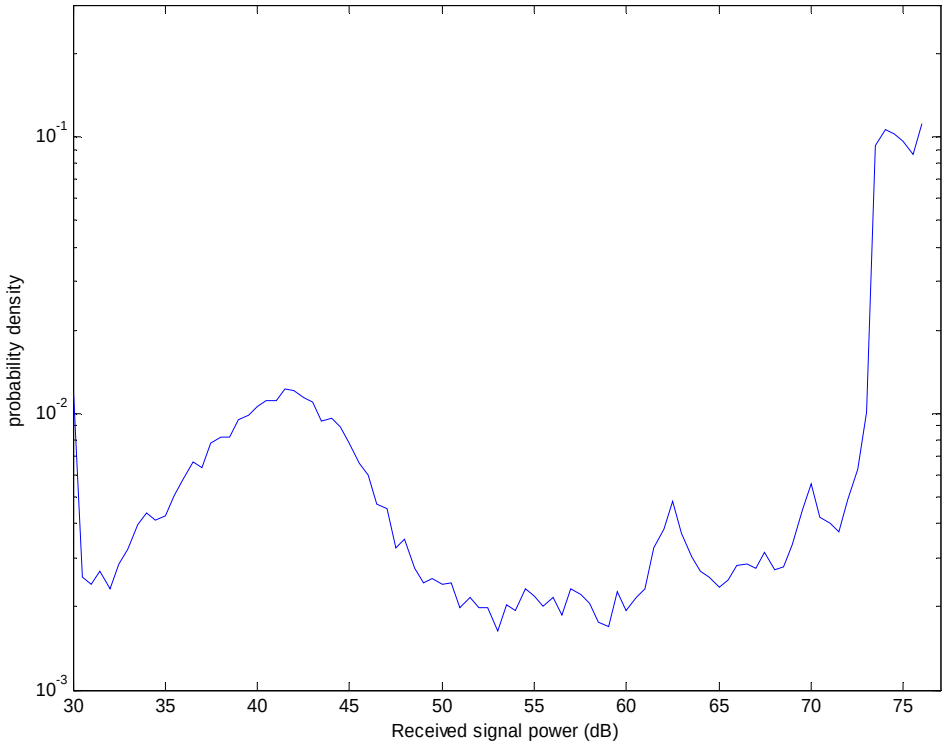

Figure 20. Histogram, spatial data (signal power vs. distance traveled, 100 bins).

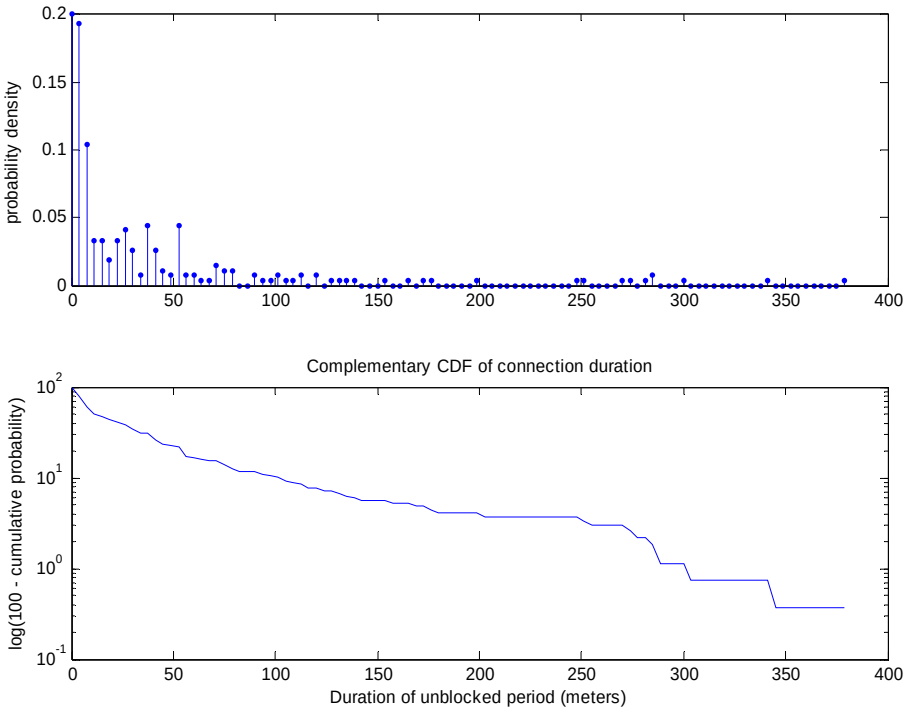

Figure 21. Histogram (top) and CDF (bottom) of connection duration (spatial data).

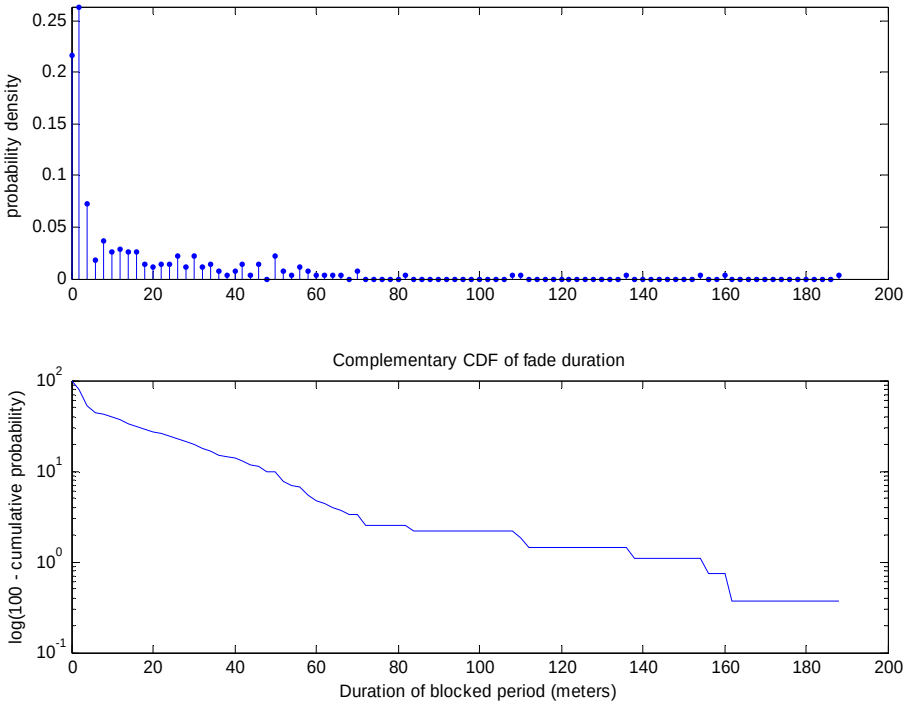

Figure 22. Histogram (top) and CDF (bottom) of fade duration (spatial data).

The histogram in Figure 20 is surprisingly similar to that computed for the temporal data. It clearly shows the strong impact and bimodal nature of the connection/fade process. Like their temporal counterparts, the complementary CDFs plotted in Figures 21 and 22 are roughly approximated as linear, which reflects a nearly geometric distribution of dwell times.

It is unfortunate that some of the empirical data sets were collected without velocity information. This exacerbates the small data set problems already introduced in Section II.A. This problem is reflected in Figures 21 and 22 which both show large drop outs in the histograms where no dwell times were collected. The reduced data set only includes 530 state transitions, which equates to 265 dwell intervals in each state.

C. MODELING THE EMPIRICAL DATA

While the first two sections of this chapter focus on understanding the available empirical data (e.g., observations) and processing that data to make it useful, this section

focuses on the value of that data for developing an accurate generative model of the urban signal fade process. This is the final step towards achieving our short term goal of developing a model of the urban signal fade phenomena. To achieve this goal we use parameter estimation to estimate accurate model parameters. Once we have an accurate model of the urban signal fade process, we can proceed with the long term goal of developing a network simulation for investigating solutions.

As discussed previously, the Markov model is a popular choice as a generative model of the LMSC. Agreeing with Schodorf, we will limit our current investigation to a two-state version [5]. The dominant effect in the urban environment is complete signal blockage by a building. As shown in Figure 13, the histogram is bimodal reflecting this dominance of two states. While other authors have investigated multi-state Markov models, they were predominantly investigating signal strength in many environments (i.e., open, rural, suburban, urban).

1. Introduction and Theory of Parameter Estimation

The problem of parameter estimation is the problem of fitting the model to the empirical data by adjusting the model parameters. For situations where analytical solutions are not viable, many disciplines apply a version of the Expectation-Maximization (EM) algorithm. EM algorithms search for the maximum likelihood model parameters of an unknown or partially unknown probabilistic process based on a set of samples from that process. The basic approach to determining the solution is by iteratively engaging in two steps, which together form the name of the algorithm. The first step is the Expectation step, which computes the expectation of the complete-data log-likelihood with respect to the unknown parameters given the observed data and current estimates of the unknown parameters [44]. The complete data set is defined to include the known data (e.g., observations) and the unknown or hidden data (model parameters). The first iteration starts with a simple guess for the unknown parameters. The second step is the Maximization step, where the value of the expectation computed in step 1 is maximized by manipulating the freshly computed "unknown" parameters.

These new parameters become the seed for the next iteration. The iterations can continue until the log-likelihood is no longer growing when compared to an acceptable tolerance ε. The EM algorithm is guaranteed to converge on the maximum likelihood solution [44]. The remainder of this section focuses on the background and implementation of the Baum-Welch Algorithm, which is a specific version of the EM algorithm developed for estimating HMM parameters [37, 38, 44, 45].

a. Nomenclature

At this point, it is important to specify the nomenclature used in the remainder of this discussion. For the most part, we have adopted the nomenclature and derivation of Liporace [46]. Some of the differences with Rabiner [30] are noted. Table 7 shows the detailed variable assignments and possible subscripts.

Table 7. Markov model and Baum-Welch nomenclature.

Variable	Name (where appropriate) and Definition
S	The number of states in the Markov model. Rabiner and Juang use "N" for this variable. S is always a positive integer.
T	The total number of time steps in the finite output sequence under analysis.
t	The current time step which increases monotonically from 0 to T.
s_t	The current state at time t. Rabiner uses "q" for this variable. The entire vector of states will be represented by dropping the subscript, $\mathbf{s} = \{s_0 \ldots s_T\}$.
O_t	Observation: The output of the model at time t. The entire vector of observations will be represented by dropping the subscript, $\mathbf{O} = \{O_0 \ldots O_T\}$.
A	The transition matrix is composed of the individual transition probabilities, a_{ij}. The matrix size is S x S.
a_{ij}	Transition probability: The probability of transitioning from state i to state j in a discrete time step: $\Pr(s_{t+1} = j \mid s_t = i)$. The value of a_{ij} is 0 to 1 inclusive. The subscripts i and j can be any number from 1 to S.
a_i	The initial state probability is the probability of being in state i at $t = 0$, $\Pr(s_0 = i)$. Rabiner uses "π" for this variable.
π_i	The steady state or stationary probabilities are the long term probabilities of being in state i. This is also the m-step transition probability at the limit: $\lim_{m\to\infty} a_{ij}^{(m)} = \pi_j$ [25]. If $a_i = \pi_i$, then the Markov process is called "stationary".
$b_j(O_t)$	The probability of the model generating output observation O_t while in state j. This parameter can be a discrete or continuous probability density function

	in vector or equation form.
λ	The set of parameters that uniquely define a hidden Markov model. The set usually includes the transition probabilities, a_{ij}, the output probabilities, $b_j(O_t)$, and the initial state probabilities, a_i.
$L(x\|y)$	The likelihood of x given y where y is assumed fixed. In maximum likelihood, we attempt to find the value of x that maximizes the expression.
$N(m,\sigma)$	A Normal or Gaussian distribution function with mean, m, and standard deviation, σ.

b. Parameter Estimation of hidden Markov models: Baum-Welch

Many different authors have identified and discussed the three basic problems associated with hidden Markov models [30, 44]. The three problems are restated in Table 8. We will focus the remainder of this discussion on Problem 3, which is also known as Parameter Estimation.

Table 8. The basic problems associated with HMMs.

Problem 1	If you know the HMM's parameters, then what is the probability of a specific vector of output data: $Pr(O\|\lambda)$?
Problem 2	If we have a vector of output data, what is the state sequence that most likely produced the output?
Problem 3	If you have output data and imperfect estimates of the model's parameters, then how can we improve the model's parameters and maximize $Pr(O\|\lambda)$?

One popular approach to solving the third problem is to use the Baum-Welch Algorithm (BWA). The Baum-Welch algorithm is a specific form of the general Expectation-Maximization (EM) algorithm specifically developed for parameter estimation of hidden Markov models [30, 44]. The remainder of this section focuses on the background and general implementation of the BWA. For this specific problem, the known data set consists of the process samples that form the observation vector. The unknown data set consists of the state transition probabilities and intrastate probability density function parameters for output transmission.

As mentioned above, the foundation of BWA (and EM) is the likelihood function. The goal is to determine the likelihood of a set of observations, $\mathbf{O} = \{O_1, \ldots,$

$O_T\}$ given estimates of the model's parameters, $\lambda = \{a_{ij}, b_j\}$ for $i, j = 1 \ldots S$. When the sequence of states is known, $\mathbf{s} = \{s_0, s_1, \ldots, s_T\}$, then the likelihood of the observation sequence is defined as

$$L_\lambda\left(\mathbf{O}|\mathbf{s}\right) = \prod_{t=1}^{T} b_{s_t}\left(O_t\right). \tag{2.3}$$

Equation (2.3) shows that the likelihood of the observation sequence **O** conditioned on the sequence of states is simply the product of the probability of each individual observation. As a simple example, let us assume that we have a two-state Markov chain where the output of each state is a Gaussian random variable. A state transition represents a shift in the mean of a Gaussian random variable. We define State 1 as N(0,1) and State 2 as N(1,1). We have an initial state ($t = 0$) plus two samples from this Markov Chain, so $T = 2$.

The observation sequence is $\mathbf{O} = \{-0.4, 1.1, 0.6\}$, and the corresponding state sequence is $\mathbf{s} = \{0, 1, 1\}$. The likelihood of this observation sequence conditioned on the state sequence is

$$L_\lambda(\mathbf{O}\,|\,\mathbf{s}) = b_{s_0}(O_0)b_{s_1}(O_1)b_{s_2}(O_2) = f_x\left(x = -0.4\,|\,s_0 = 0\right) f_x\left(x = 1.1\,|\,s_1 = 1\right) f_x\left(x = 0.6\,|\,s_2 = 1\right)$$

$$= \frac{1}{\sqrt{2\pi}} e^{-\frac{(-0.4-0)^2}{2}} \frac{1}{\sqrt{2\pi}} e^{-\frac{(1.1-1)^2}{2}} \frac{1}{\sqrt{2\pi}} e^{-\frac{(0.6-1)^2}{2}} = \left(0.3683\right)\left(0.3970\right)\left(0.3683\right) = 0.2924.$$

Now let us compare this result with a situation where the states are reversed to $s_0 = 1$, $s_1 = 0$, $s_2 = 0$. The new likelihood is

$$L_\lambda(\mathbf{O}\,|\,\mathbf{s}) = b_{s_0}(O_0)b_{s_1}(O_1)b_{s_2}(O_2) = f_x\left(x = -0.4\,|\,s_0 = 1\right) f_x\left(x = 1.1\,|\,s_1 = 0\right) f_x\left(x = 0.6\,|\,s_2 = 0\right)$$

$$= \frac{1}{\sqrt{2\pi}} e^{-\frac{(-0.4-1)^2}{2}} \frac{1}{\sqrt{2\pi}} e^{-\frac{(1.1-0)^2}{2}} \frac{1}{\sqrt{2\pi}} e^{-\frac{(0.6-0)^2}{2}} = \left(0.1497\right)\left(0.2179\right)\left(0.3332\right) = 0.0109.$$

Thus, the likelihood of receiving the observation sequence, $O = (-0.4, 1.1, 0.6)$, is significantly lower if the state sequence is $s = (1, 0, 0)$.

The purpose of the λ subscript found in the expression for the likelihood L_λ is to show the additional dependence of the likelihood on the model parameters. For this specific example, our choice of mean and variance for each Gaussian probability function had a significant impact. The likelihood function can also be dependent on the transition probabilities a_{ij}.

While Equation (2.3) shows the conditional likelihood, we are actually interested in the posterior likelihood $L_\lambda(O)$ not conditioned on the state sequence (since we will not usually know the state sequence). In order to remove the conditional dependency, we can build a weighted average of the conditional likelihoods for each possible state sequence. The weighting is based on the probability of that particular state sequence occurring, which we can label $P_\lambda(s)$ as the probability of the state sequence **s** given the model parameters $\lambda = \{a_{ij}, m_i, \sigma_i\}$:

$$P_\lambda(s) = a_{s_0} \prod_{t=1}^{T} a_{s_{t-1}s_t}. \tag{2.4}$$

Equation (2.4) shows that P_λ is easily computed as the probability of starting in state s_0 multiplied by the probability of each transition in the sequence. To continue our simple example, assume that $a_0 = a_1 = 0.5$ and $A = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$. We can compute the probability of the first state sequence **s** = {0, 1, 1} as $P_\lambda(s) = a_0\, a_{01} a_{11} = (0.5)(0.2)(0.7) = 0.07$. Similarly, the probability of the second state sequence **s** = {1, 0, 0} is $P_\lambda(s) = a_1\, a_{10} a_{00} = (0.5)(0.3)(0.8) = 0.12$.

Combining, Equations (2.3) and (2.4), we can finally compute the likelihood of an observation sequence given the estimated model parameters:

$$L_\lambda(\mathbf{O}) = \sum_s L_\lambda(\mathbf{O}|\mathbf{s}) P_\lambda(\mathbf{s}) = \sum_s a_{s_0} \prod_{t=1}^{T} a_{s_{t-1}s_t} b_{s_t}(O_t). \tag{2.5}$$

The goal is to maximize the posterior likelihood $L_\lambda(\mathbf{O})$ to find the optimal set of parameters λ. Continuing our example, with $S = 2$ and $T = 2$, we see that the summation

over the state vector expands into eight individual terms representing all possible sequences of length three ($S^T = 2^3 = 8$).

The likelihood function can be viewed as a complex surface in an N-dimensional space where N is the number of parameters in λ. From this perspective, maximizing the likelihood function by adjusting the model parameters is simply finding the "highest" point in the complex surface. This problem does not yet have an analytical solution. Instead, Baum, Petrie and others derived a specific version of the Expectation-Maximization algorithm to solve the problem iteratively. A complete discussion of the derivation and iteration details can be found in [31, 37, 38, 45, 46]. The basic steps are summarized here:

1. Collect observation vector **O** from stochastic process.
2. Assume a discrete number of states for the process.
3. Make initial guess for unknown parameters (i.e., a_{ij}, μ, σ).
4. Compute $\sum_{t=1}^{T} \gamma_i(t)$, which is the expected number of times the process is in state *i* for the observation sequence of duration *T*. It is important to note that this expression also represents the number of times the process leaves state *i*.
5. Compute $\sum_{t=1}^{T-1} \varepsilon_{ij}(t)$, which is the expected number of transitions from state *i* to state *j* for the observation sequence of duration *T*.
6. Realizing that the transition probabilities a_{ij} are simply a ratio of the expected number of transitions from state *i* to state *j* to the total number of transitions leaving state *i*, we can update our estimate of a_{ij} by dividing the result from step 5 by the result from step 4.
7. Update the output transmission function parameters:

    7a. For discrete output transmission functions (e.g., Gilbert model), we can compute the transmission probability by taking a ratio of the

expected number of times the observation had a value of v_k while in state i compared to the total number of times in state i:

$$\tilde{b}_i(k) = \frac{\sum_{t=1}^{T} \gamma_i(t)\,\delta_{o_t, v_k}}{\sum_{t=1}^{T} \gamma_i(t)}, \tag{2.6}$$

where $\delta_{x,y}$ represents a function that equals one when $x = y$.

7b. Parameter update equations are more complex for continuous transmission functions. Relevant equations can be found in [44] and [46] for mixed Gaussian transmission probability distributions which require the update of mean μ and variance σ^2.

2. Implementing Baum-Welch

As mentioned in Section I.D, in [5] Schodorf attempted to model the urban fade channel with two-state generative models, using both non-hidden Markov model and Gilbert model approaches. Each time the two states are identified as *good* and *bad*, which are the same as connection and fade. The Gilbert model is introduced in Section I.C. as the simplest version of a hidden Markov model. The *good* state is completely discrete and not hidden by any stochastic output transmission function (*observable*). But the *bad* state is defined to have an internal Bernoulli transmission function that allows it to produce both connection and fade levels of performance. This small change hides the true state from direct observation. All of these models are simple enough that they do not require an Expectation-Maximization algorithm like Baum-Welch [5], as model parameters can be directly estimated from the empirical data set. Schodorf's results are summarized in Section 3 where it can be seen that additional improvements appear to be possible. In an effort to achieve those additional improvements, we increase model complexity by assuming continuous output transmission functions.

We tried three additional hidden Markov models, each with a continuous density function for the output transmission. In the first approach, we assumed a single Gaussian distribution for the output of each state, which is similar to the example discussed in the

last section. The second approach models each state as a weighted mixture of five Gaussians. The additional complexity of this approach allows the algorithm to resolve a more complex transmission function. The final approach converts the histograms presented earlier in this chapter into continuous pdfs by implementing a piecewise polynomial. The benefits of this last approach are a substantial reduction in unknown parameters and a more accurate continuous transmission function.

a. Transmission Function: Gaussian or Gaussian Mixture

The smallest increase in complexity that still provides a continuous probabilistic output transmission function is a single Gaussian output function. The unknown parameters are the transition probabilities and the mean and variance for the Gaussian pdf associated with each state (i.e., a_{00}, a_{11}, μ_0, μ_1, σ_0^2, σ_1^2). This makes a total of six unknown parameters to be estimated, which is double the three necessary for a Gilbert model (i.e., a_{00}, a_{11}, h).

While a single Gaussian transmission pdf may provide a more accurate representation of the signal strength than the Bernoulli function used by Schodorf, it is still a poor match to the histograms shown in Figures 10 and 13. A mixture of multiple Gaussians can often be an increasingly accurate replacement for a non-Gaussian stochastic process. The first and only attempt at this approach involved a five Gaussian mixture for each state. The results of this approach were promising and are found in Section 3.b. The 32 unknown parameters include two transition probabilities, the mean and variance for ten Gaussian distributions and ten weighting factors. As shown in the results, the fourth and fifth mixture components were minor contributors and an increase in the mixture count was not necessary.

The implementation of this approach relied heavily on the work of Kevin Murphy [47]. Instead of developing new Matlab scripts from scratch, we started with Murphy's Bayesian Toolbox which includes several hidden Markov model related processes including parameter estimation.

b. Transmission Function: Hermite Interpolant

While a mixture of weighted Gaussian functions provides a significant amount of flexibility in representing a complex stochastic process, it may not be as accurate as using the actual data itself. One idea for achieving these improvements was to model the process output with a more accurate probability density function. Thus, we investigated deriving an analytical expression for the histogram (as shown in Figures 10 and 13) using a splined or hermite interpolant piecewise polynomial. The difference between a hermite interpolant and a spline interpolant is the additional constraint that spline interpolants be continuous in the second derivative when transitioning between polynomials. Both hermite and spline interpolants are required to be continuous in the first derivative (slope). Both approaches were applied to the data, and the differences were very small. The splined interpolant have more trouble with the sharp transitions of the histogram, so for the remainder of this discussion we focus on the hermite interpolant.

The first step towards completing this approach is to split the histogram into two distinct pdfs, each representing a state. Based on the concept that the two states are each represented by a peak in the histogram, the logical separation point is the low point between those two peaks. A windowed average of the histogram showed that the general low point between the two peaks is between 59 and 60 dB. However, it is not feasible to simply split the histogram into two pdfs, one below 59 dB and one above 59 dB. This would make the Baum-Welch algorithm unnecessary because there would no longer be any uncertainty about which state the process was in. Instead, it is imperative that the pdf for each state cover the full range of possible output values. To accomplish this separation, the histogram value at 59 dB was shared equally with both pdfs. Then a straight line was formed from that center point to the distant edge of the histogram. Each straight line was sloped to gently decrease until reaching zero at the edge of the pdf.

The resulting analytical pdfs are shown graphically in Figures 23 and 24. These can be compared to Figures 10 and 13, which shows the original data and smoothed data histograms.

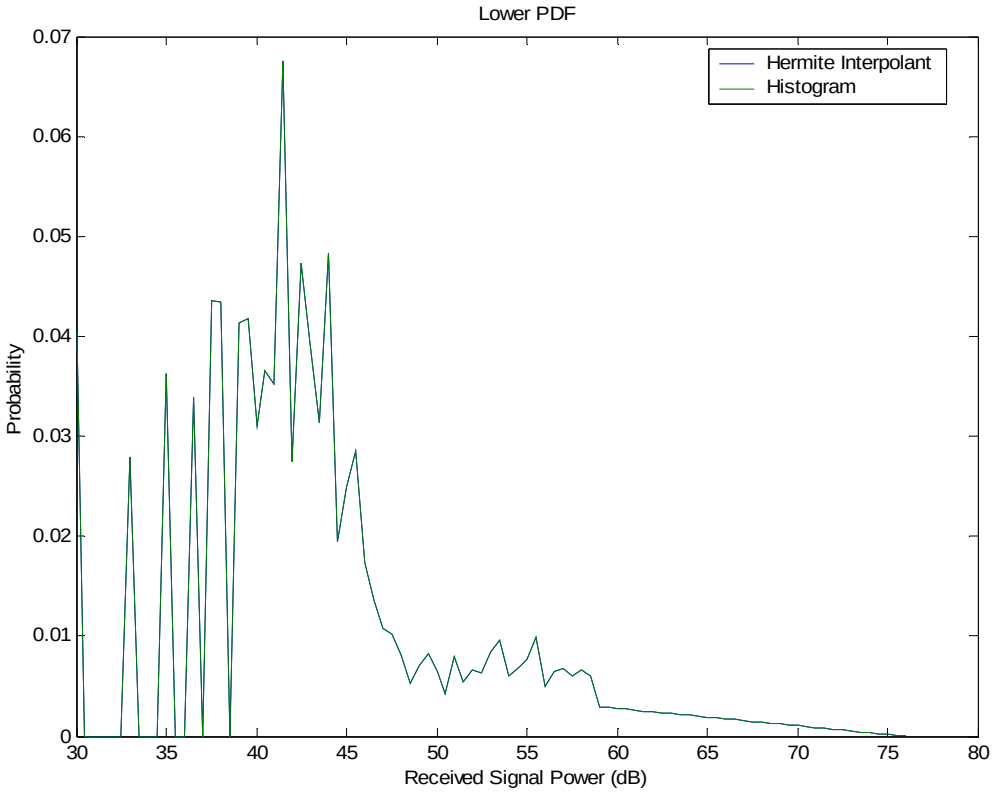

Figure 23. Piecewise polynomial pdf of State 0 (fade).

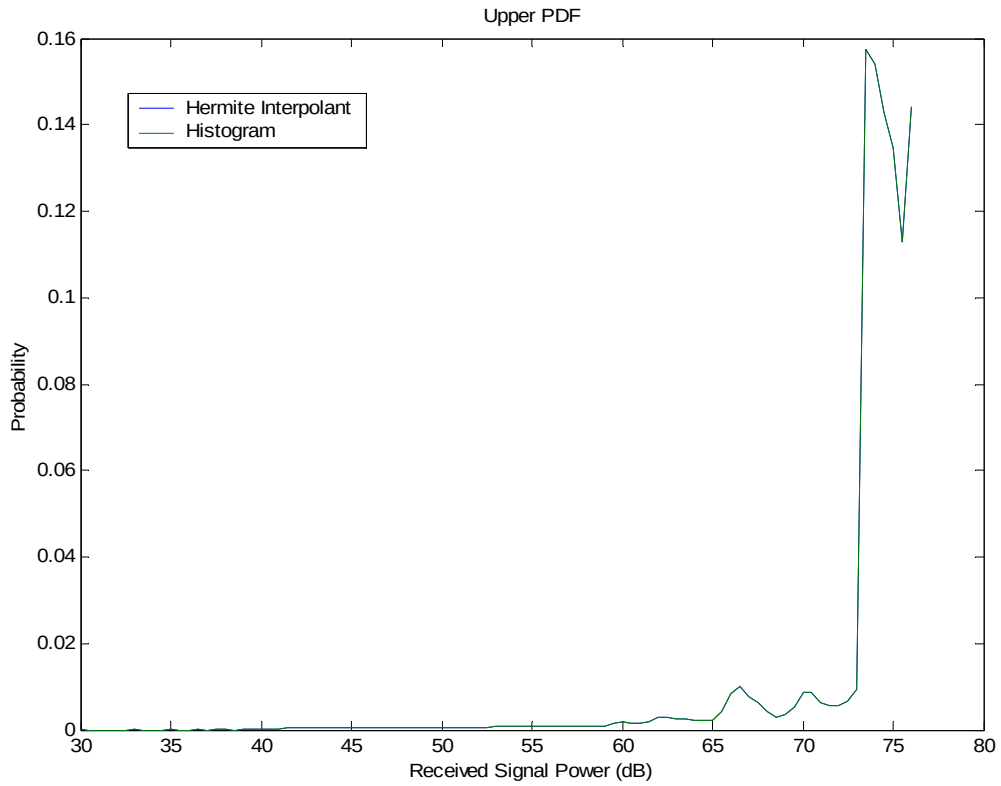

Figure 24. Piecewise polynomial pdf of State 1 (connection).

One important impact of this approach was the simplification of the Baum-Welch parameter estimation. When modeling the process with a mixture of Gaussian random variables, we have to iteratively estimate the mean, variance and weight coefficient of each Gaussian. With this piecewise polynomial approach, the pdf is assumed to be known and is no longer updated iteratively. In theory, the pdf is assumed to be nearly perfect and does not require any modification. The simplification provides a significant reduction in required mathematical operations and computational run-time.

c. Initial Guess for Unknown Parameters

As presented earlier in Section C.1.b, Step 3 of the BWA Expectation-Maximization process determines an initial guess for the unknown parameters, where unknown parameters are those we are trying to estimate. The EM process computes a likelihood of these parameters having those initial values and then attempts to change the

parameters in a direction that increases the likelihood. As part of exploring the performance of the BWA in this context and to understand the sensitivity to the initial guess, we repeated the algorithm multiple times with either high confidence *good* initial guesses or randomly generated initial guesses. Table 9 shows the three different approaches that were used to generate an initial guess for unknown parameters. The initial attempt was based on the broad probability functions shown in the row labeled "Random, Broad". This approach had some problems and led to the development of the "Random, Focused" input parameters. The constraints on this set of initial guesses are reasonable assumptions based on the amount of information available from the empirical data analysis. Finally, an attempt was made to seed the EM algorithm with a nearly ideal initial guess. The parameters shown in the row labeled "Expected Values" reflect this initial guess. The results for each approach are discussed more in Section 3. The BWA was executed repeatedly with independent random inputs and statistical information was collected about its performance.

Table 9. BWA input: variations of initial guess.

			State 1 (connection)		State 0 (fade)	
	a_{00}	a_{11}	Means	Variance	Means	Variance
Random, Focused	Uniform (0-1)	Uniform (0-1)	Uniform (65-75)	Uniform (1-3)	Uniform (30-50)	Uniform (1-3)
Random, Broad	Uniform (0-1)	Uniform (0-1)	Uniform (0-80)	Uniform (0-10)	Uniform (0-80)	Uniform (0-10)
Expected Values[1]	0.75	0.75	75, 74, 73, 71, 70	2, 2, 2, 2, 2	45, 40, 35, 30, 20	20, 20, 20, 20, 20

[1] Note: These input parameters were only applied to the weighted mixture of five Gaussian transmission functions.

3. Parameter Estimation Results

a. Schodorf's Results

For comparison purposes, a limited summary of Schodorf's results are presented here [5]. Recall from the earlier discussions in Sections I.D and II.C.2 that Schodorf tried three different approaches. The first was a non-hidden, discrete, two-state Markov model. The second and third were both implementations of the Gilbert model. The difference between his second and third modeling attempt was the approach to computing model parameters. The second model was parameterized using only the average connection statistics found in the empirical data. While this version produced an excellent fit to the empirical connection statistics, it did not provide an acceptable fit to the fade statistics. The third and final attempt was parameterized jointly using both connection and fade statistics. It is important to note that Schodorf applied these models to open, rural, and urban environments. He also ran the models with both a −3 dB connection/fade threshold and a −10 dB connection/fade threshold where the threshold is relative to LOS. We will only present the data for the urban environment with a −10 dB threshold because this is the closest match for comparison. Table 10 shows the state transition and steady state probabilities $\boldsymbol{\pi}$ estimated by each approach. Schodorf also solves for other parameters like the Bernoulli distribution parameter h but they are not repeated here since they are not valuable for comparison. Figures 6(c) and 6(d) show the complementary cumulative distribution functions of the connection and fade durations which provide a valuable check on model compatibility to the empirical data. It is important to note that Schodorf was forced to investigate the impact of setting a binary connection/fade threshold. This is a side effect of using the simple non-hidden Markov and Gilbert models which have clearly defined discrete state definitions. This step is not necessary for the continuous stochastic output functions that we investigated.

Table 10. Parameter estimation results from Schodorf [5].

	$a_{00\ (\text{connection})}$ [1]	$a_{11\ (\text{fade})}$ [1]	π_{fade} [1]
Markov (observable)	0.9726	0.9465	0.34
Gilbert	0.9748	0.9151	0.04
Gilbert (jointly opt.)	0.9959	0.9264	0.05

[1] Note: We can find the remainder of parameters by ($a_{01} = 1 - a_{00}$), ($a_{10} = 1 - a_{11}$), and ($\pi_{\text{connection}} = 1 - \pi_{\text{fade}}$)

While the Gilbert model provides improved accuracy for the open and rural environments, it does not do so for the urban environment. In [5], Schodorf selects the simple Markov model as the best overall solution for the urban COTM channel. The connection-based Gilbert model has the best fit for connection dwell time statistics, but otherwise, both Gilbert models significantly underestimate the fade statistics (a_{11}, π_{fade}). Based on the plots in Figure 6, it does appear that the Markov parameters shown above may be underestimated relative to the dwell durations. Figure 6(c) shows that short connection durations are substantially underestimated and long connection durations are overestimated. Figure 6(d) shows an opposite skew of overestimating short fades and underestimating long fades. While the Gilbert model shows a 0.1% probability of a fade exceeding 90 meters, the experimental data shows a 0.1% probability of the fade exceeding 175 meters. Further analysis of these two figures and the parameter estimates reported for Schodorf's three different attempts suggest that p_{gg} should be approximately 0.98 and p_{bb} should approximately 0.95.

b. Mixed-Gaussian Results

In an effort to find a better approach to converting empirical data into an accurate fade model, we applied the BWA avoided by Schodorf. A short introduction to parameter estimation and the Baum-Welch Algorithm were presented in Section C.1. The empirical input data was processed as discussed in Sections II.A and II.B. One of the benefits of the BWA is that you do not need to know anything about the output transmission functions. Instead, the descriptive parameters of the transmission pdfs can become additional variables estimated by the parameter estimation algorithm. A

common implementation involves assuming the transmission pdfs are a weighted mixture of individual Gaussian pdfs. In this implementation, the BWA estimates the transition probabilities of the Hidden Markovian process and the mean, variance and weighting coefficients of the combined Gaussian pdfs that represent each state. We investigated the simplest version with a single Gaussian pdf representing each state and also a more complex implementation where each state was represented by a mixture of five Gaussian pdfs. Each approach was attempted numerous times, each with a different random initial guess. The initial guess was originally expected to have only a small impact of results [44]. Some variance is expected because we are solving the iterative estimation process to a finite tolerance. This means that the true maximum has not actually been found, but we are in the vicinity. But with a tight convergence limit, like $\varepsilon = 10^{-6}$, these differences were expected to be small. Results did not completely conform to this expectation. The most noticeable impact of changing the initial guess was the number of iterations required by the BWA to converge within the tolerance.

The results for a single Gaussian transmission function and the "Random, Focused" initial guess inputs from Table 9 are shown in Table 11. The table shows numerical results for transition probabilities (a_{ii}) and transmission function parameters (μ_i, σ_i^2). As mentioned earlier, some attempts were made to start with even more divergent values for the initial input, but the algorithm was not tolerant to converging on reasonable parameter estimates with the extreme inputs labeled "Random, Broad" in Table 9. An example of this difficulty is provided in the mixed Gaussian results. Shown after each parameter is the approximate variance computed from ten independent attempts. As mentioned above, these variances are extremely tight and reflect an insensitivity to the initial guesses within the constraints shown.

Table 11. BWA results for a Gaussian transmission function (mean, variance for 10 attempts)

	Original Data (temporal)	Smoothed Data (temporal)	Spatial Data
a_{00}	0.9630, 10^{-15}	0.9670, 10^{-15}	0.9876, 10^{-15}
a_{11}	0.9820, 10^{-16}	0.9838, 10^{-16}	0.9923, 10^{-17}
State 1: Mean	74.694, 10^{-12}	74.658, 10^{-12}	74.711, 10^{-11}
State 1: Variance	1.4042, 10^{-11}	0.9520, 10^{-11}	0.9679, 10^{-10}
State 0: Mean	49.065, 10^{-10}	49.841, 10^{-10}	46.925, 10^{-9}
State 0: Variance	179.91, 10^{-8}	153.18, 10^{-7}	134.30, 10^{-6}

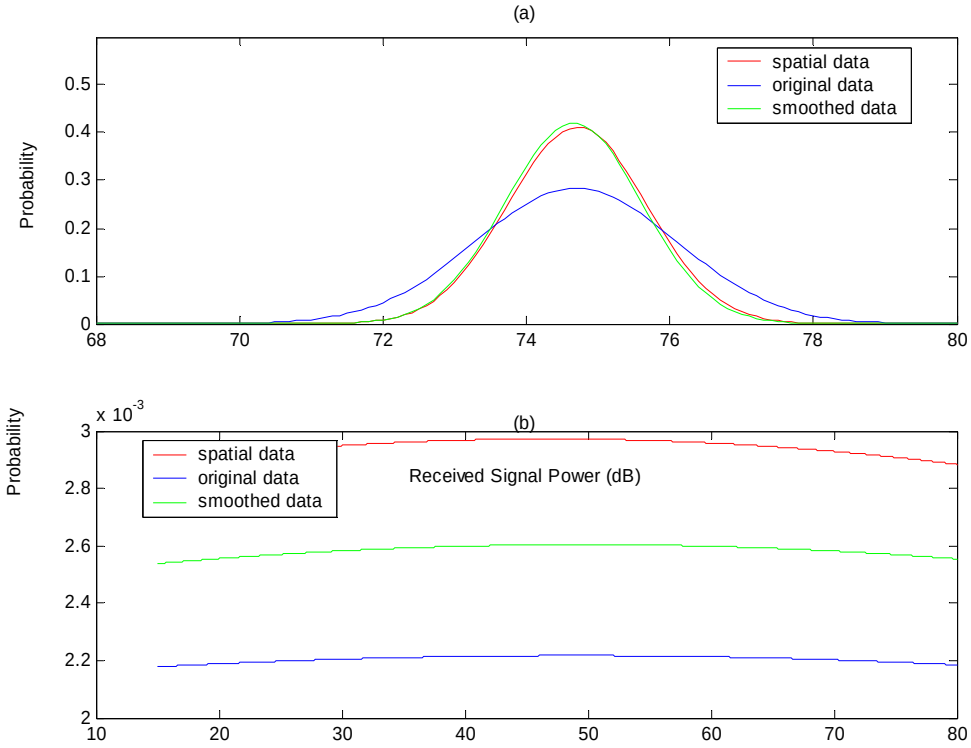

Figure 25. BWA estimated Gaussian pdfs to represent (a) State 0 (fade) and (b) State 1 (connection) for three input data sets. Horizontal axis is received signal strength measured in dB.

The parameter estimation process was repeated for each of the data sets discussed earlier in Chapter II. The spatial data (signal strength vs. meters traveled) is the appropriate data set for comparison with Schodorf's results since he used spatial data

to generate the results shown in Figure 6 and Table 10. Figure 25 shows three independent attempts to estimate the Gaussian transmission pdfs from the original data set. Comparison with the empirical histogram shown in Figure 10 shows that the similarities are minimal. The mean of the two intrastate pdfs appear in the same location as the histogram, but the additional structure of the histogram cannot be recreated by a single Gaussian. In general it appears that this approach may slightly overestimate the fade and connection statistics but appears to be more accurate than the Gilbert model used by Schodorf. It is important to note that the transition parameters were larger for the smoothed data than for the original data set. This behavior is appropriate since the smoothed data has a reduced number of state transitions due to the removal of single time-step signal fades.

A more advanced approach to representing complicated transmission functions is to model them as a weighted mixture of multiple Gaussian functions. The BWA results for a weighted mixture of five Gaussian functions are shown in Tables 12 and 13 and Figures 26 through 32. Table 12 and Figures 26 and 27 show the results of a single parameter estimation with an initial guess that was very close to expectations, which is shown in Table 9 as "Expected Values". Figure 26 shows the five individual mixture component pdfs separately but modified by their weighting coefficients. In State 1 (connection), the weighting coefficients for pdfs 4 and 5 are so small as to make those mixture components nearly negligible. Figure 27 presents the final output transmission functions $b_1(O_t)$ and $b_2(O_t)$, which are computed by combining the mixture components in Figure 26.

Table 12. Estimated model parameters for transmission functions.

		Gaussian #1	Gaussian #2	Gaussian #3	Gaussian #4	Gaussian #5
State 1	Means	75.523	74.721	73.669	71.301	67.936
	Variances	0.669	1.291	0.141	3.793	6.343
	Mixture weights	0.4154	0.1811	0.3012	0.0276	0.0747
State 0	Means	55.036	42.797	38.026	31.702	19.741
	Variances	62.248	13.673	25.8977	4.651	0.010
	Mixture weights	0.3440	0.3271	0.2752	0.0062	0.0474

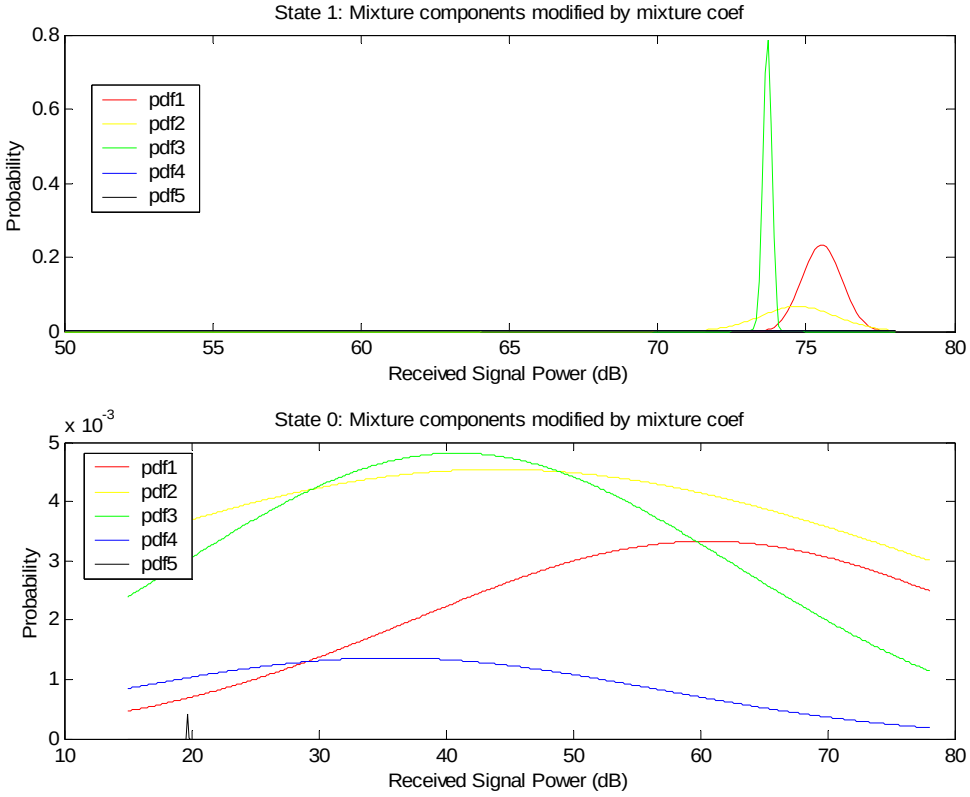

Figure 26. BWA estimated mixture of five Gaussian pdfs each to represent States 0 (fade) and 1 (connection); each shown separately.

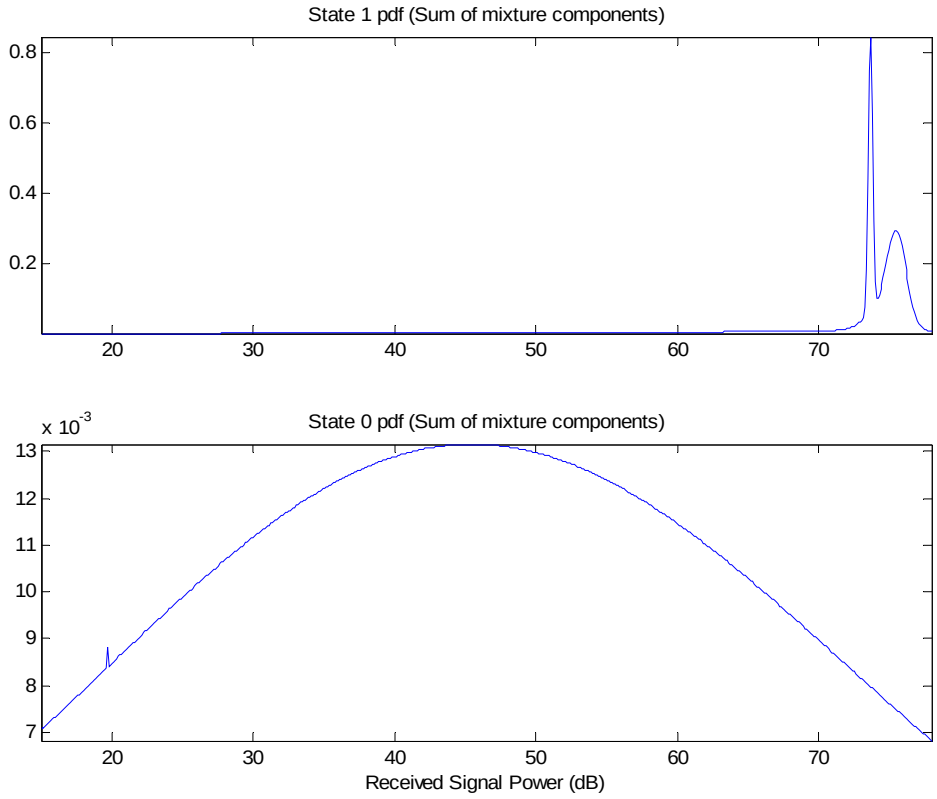

Figure 27. BWA estimated mixture of five Gaussian pdfs each to represent States 0 and 1; each mixture combined to form a single pdf.

Like the single Gaussian approach, the Gaussian mixture approach was also attempted with random initial guesses for the desired parameters (see Table 9). The performance of the BWA varied noticeably with initial guess as shown in Figures 28 and 29. Figure 28 shows the transmission pdfs estimated by the algorithm for five independent attempts. Figure 29 shows the log-likelihood as a function of algorithm iteration for those same five attempts. The *y*-axis of Figure 29 is zoomed into a tight range of log-likelihood to see the slight changes that occur at higher iterations, but this cuts the first one or two iterations off. It is not uncommon for the BWA to start with a log-likelihood of less than −130,000 with the "Random, focused" initial guess constraints shown in Table 9. As shown in Figure 28, these different attempts do not converge to the same transmission function within 200 iterations of the Expectation-Maximization cycle.

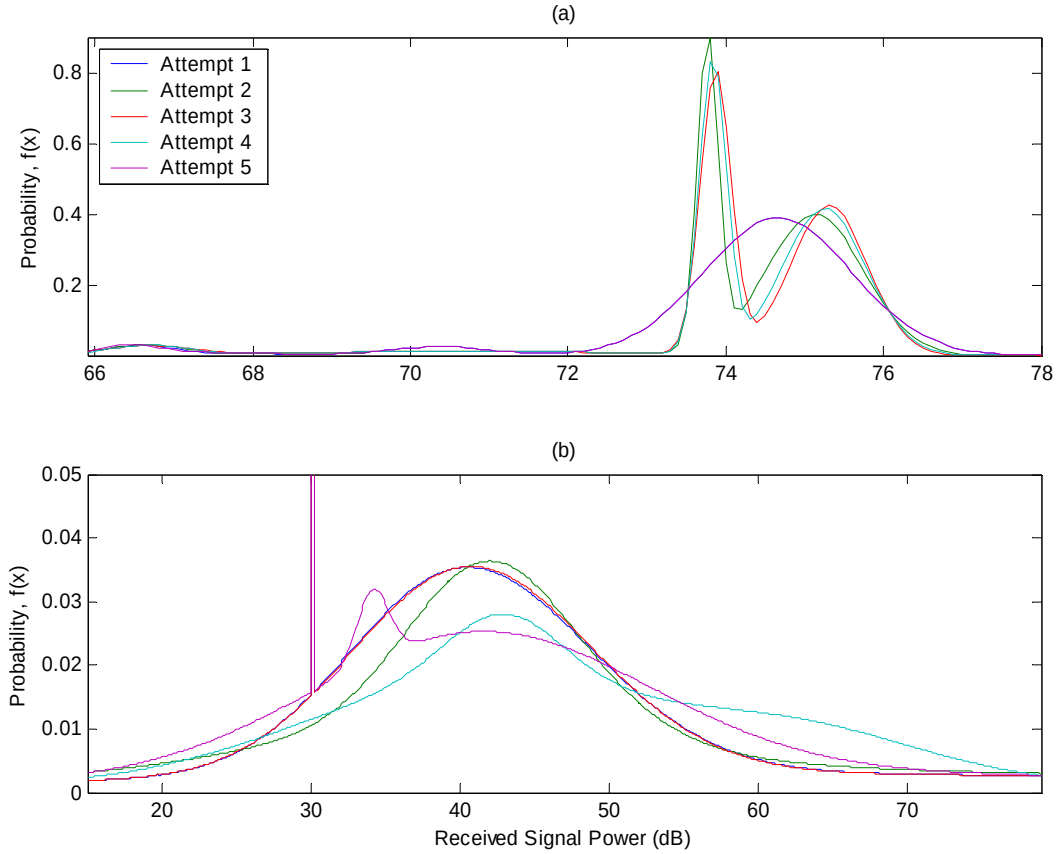

Figure 28. Output transmission pdfs for five parameter estimation attempts using the original data set; each with a different random initial guess with "Random, focused" constraints shown in Table 8. (a) State 1 (connection), (b) State 0 (fade).

For State 1, attempts 1 and 5 resolved only one peak in the transmission function, while the remaining attempts resolved two peaks. Please note that the plot for attempt 1 is difficult to see because it is hidden behind attempt 5. It is interesting to note that the single peak estimates have essentially the same mean as the single Gaussian approach in Figure 25. For State 0, attempts 1, 3 and 5 resolved a distinct and heavily weighted spike in the pdf near 30 dB of signal strength. This is surprising because the empirical histograms do not show a similar structure. Attempts 1 and 5 certainly stand out as generally being of lower accuracy. It is not surprising to see that they have the lowest log-likelihoods of the five attempts. It is mildly surprising that attempt 3 has the

inaccurate State 0 pdf especially since its log-likelihood is the highest by far, but it was not finished iterating when it reached the limit of 200 iterations. If allowed to continue, it may have shifted the estimate to be a closer match to expectations. The remaining five iterations (not shown) were similar to the examples presented. The majority of attempts resolved two peaks in the State 1 pdf and did not resolve the sharp spike in the State 0 pdf.

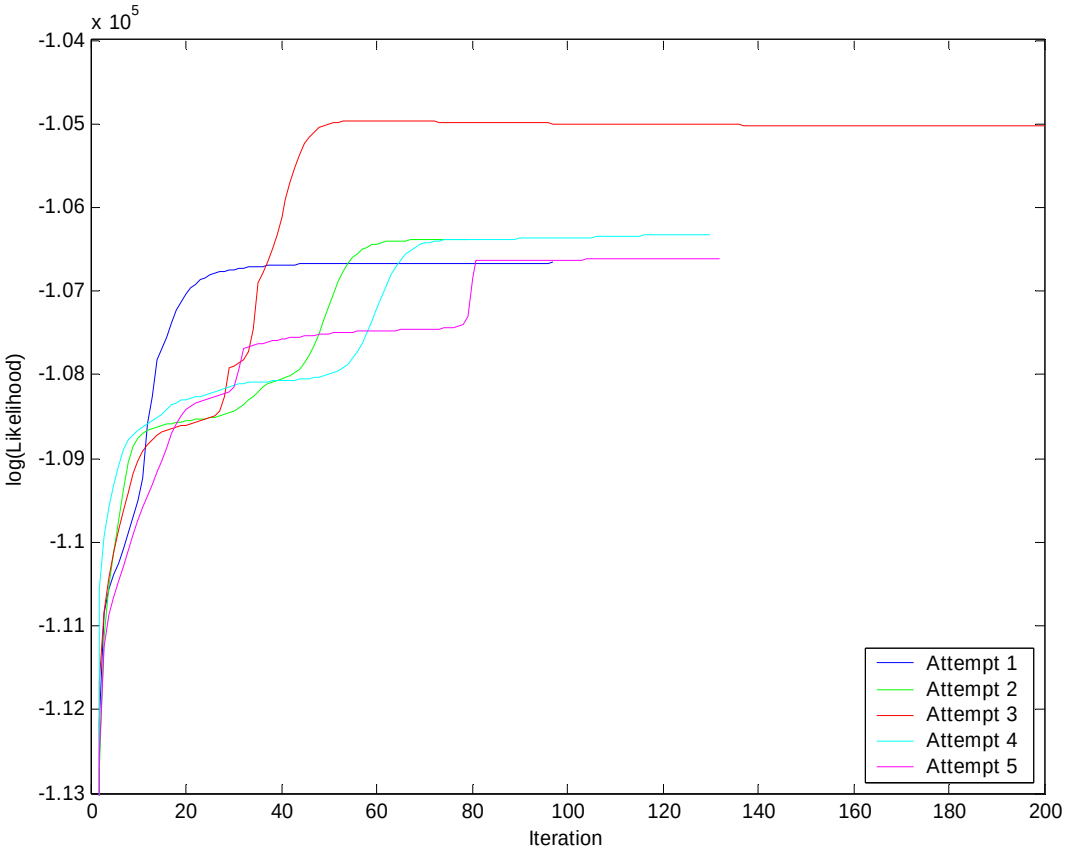

Figure 29. Likelihood for five separate attempts, five different initial guesses, original data.

This process was repeated with the two additional data sets: smoothed and spatial. The results for transition probability are summarized in Table 13. Figures 30 and 31 are similar to Figures 28 and 29, respectively, except that they are based on the spatial data set. An important difference between the spatial data set and the temporal data set is

the number of iterations necessary to prosecute the BWA. While the BWA was frequently able to determine the maximum-likelihood solution within 150 iterations for the original and smoothed data, it in many cases required more than 250 iterations to accomplish the same with the spatial data. As can be seen in Figure 30, even after 250 iterations, two of the five attempts (1 and 4) were still resolving on a single peak. It is also important to realize that these two attempts show the lowest likelihoods in Figure 31. It is also interesting to note that the likelihoods based on spatial data were all considerably lower than those achieved with the original data set (shown in Figure 29).

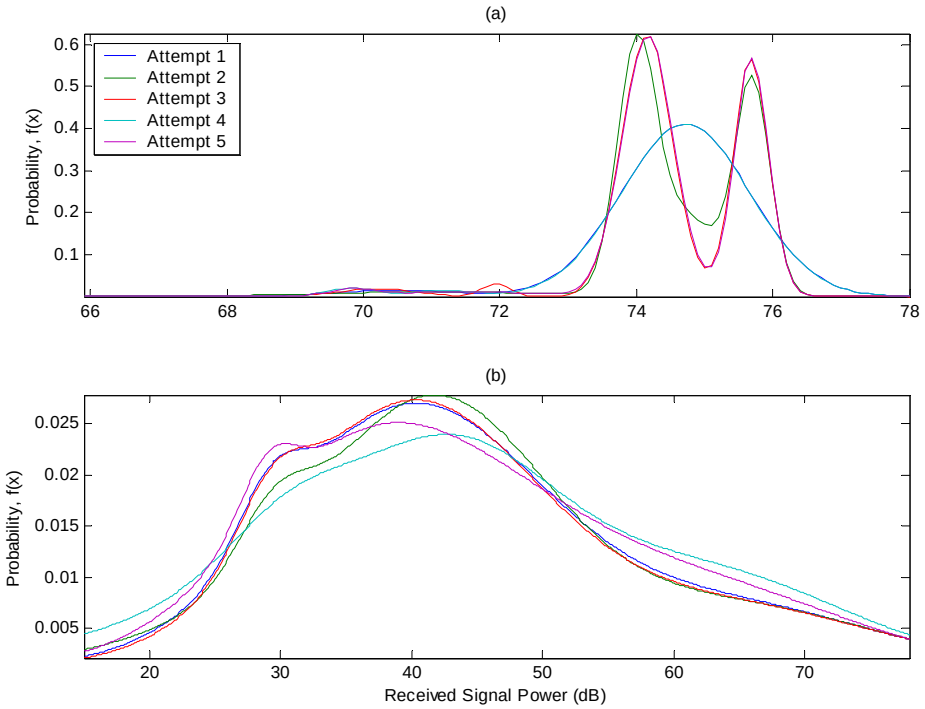

Figure 30. Output transmission pdfs for five parameter estimation attempts using the spatial data set; each with a different random initial guess with "Random, focused" constaints shown in Table 8. (a) State 1 (connection), (b) State 0 (fade).

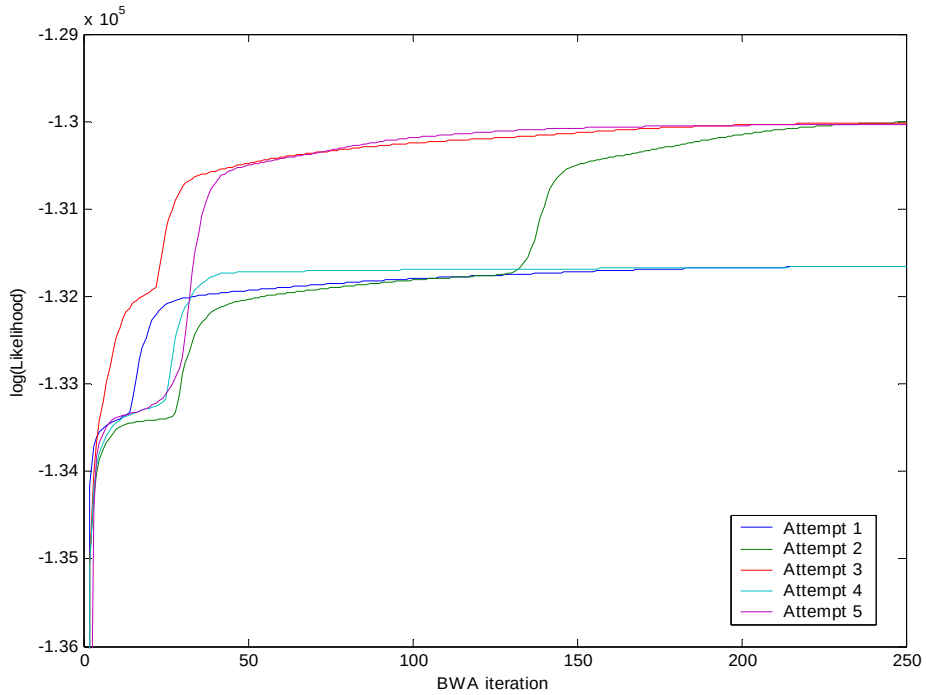

Figure 31. Log-likelihood for the five BWA attempts shown in Fig 25 (spatial data).

The estimates of transition probability are shown in Table 13. By comparing these results to those in Table 11, we can see that the increased number of Gaussian pdfs used to represent the intra-state emission process unanimously increases the estimates for transmission probability a_{ii}. Table 13 also shows the increase in transition probability expected as a result of smoothing.

Table 13. BWA results: Gaussian mixture transition probabilities (mean, variance of 10 trials).

	a_{00}	a_{11}
Original Data	0.9657, 3E−6	0.9867, 2E−6
Smoothed Data	0.9677, 1E−9	0.9881, 5E−10
Spatial Data	0.9884, 3E−8	0.9935, 5E−8

As mentioned earlier in this section, the algorithm convergence was sensitive to the initial guess. The initial approach to randomly generate inputs, labeled

"Random, Broad" in Table 9, produced widely varying results and in some cases mixed States 0 and 1. Four independent examples of this problem are shown in Figure 32. These attempts did not reach convergence within 200 iterations so additional improvements may have continued to be manifested if additional iterations were allowed.

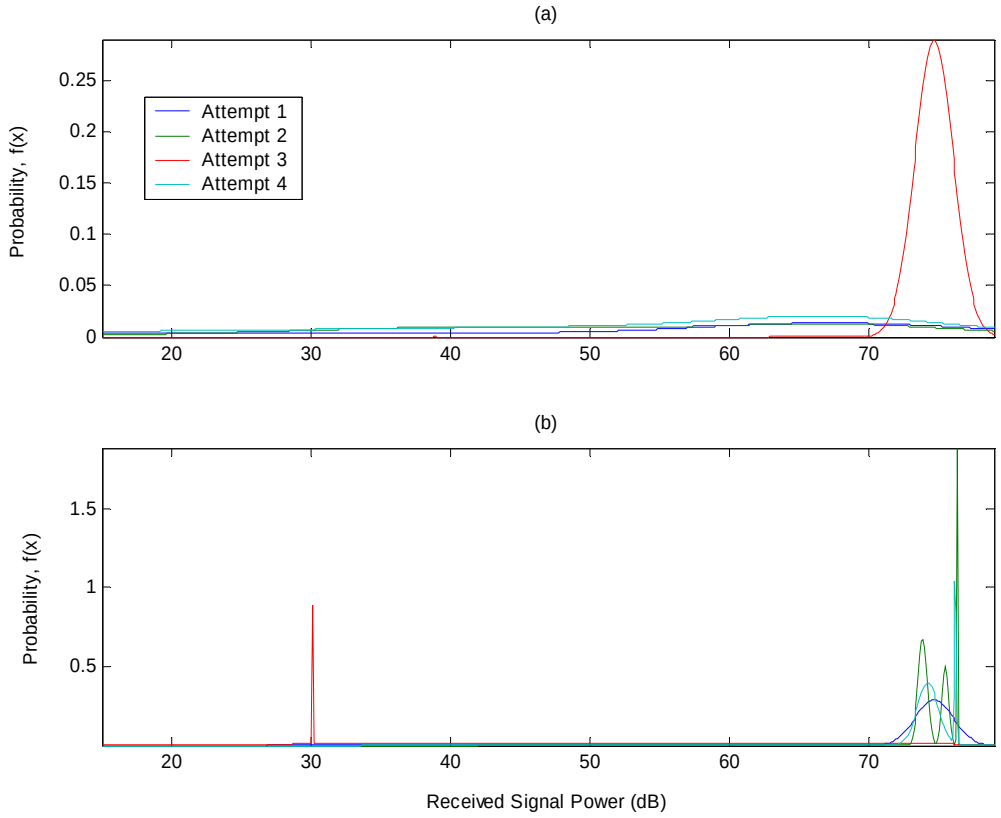

Figure 32. Output transmission pdf results for broad random initial guess (original data set).

c. Specific Discussion of Piecewise Polynomial Results

While the results from modeling the output of each state as a mixture of Gaussian random variables were promising, additional improvements seemed possible with a more accurate output transmission function. For the third and final approach, the

weighted Gaussian mixtures presented in the last section were replaced with the hermite interpolant-based, continuous pdfs shown in Figures 23 and 24 and discussed in Section 2.c.

As noted above, the number of unknown variables drops significantly with this approach because the output transmission pdfs no longer have parameters to be estimated. The only remaining parameters left for the Baum-Welch algorithm to estimate were the model's transition probabilities a_{ij}. The algorithm was applied to both the original data set and the smoothed data set to understand the impact of data smoothing. The initial guess for a_{ii} was 0.75 for the implementation shown. This matches row 3 of Table 9 and has been referred to as the "Expected Values" initial guess. Like the single Gaussian, other initial guesses were used without a significant difference in results. The final parameter estimations for a_{ii} are shown in Table 14. It is important to note that the smoothed data had larger transition probabilities than the unsmoothed data. This is the correct direction of change for these parameters since smoothing removes very short duration transitions.

Table 14. BWA estimates of transition probabilities using Hermite Interpolant.

	a_{00}	a_{11}
Original data	0.9906	0.9713
Smoothed data	0.9910	0.9721

These transition probability results appear to be slightly high relative to the interarrival times shown in the empirical data. Focusing on the results for the smoothed input data, we see in Table 5 shows a directly computed mean connection time of 7.92 seconds. Yet a transition probability of 0.99095 estimates a mean connection time of 11.05 seconds (see Equation (2.7)). Likewise, the computed mean fade duration is 2.44 seconds, but the transition probability of a_{11} = 0.9721 estimates a mean of 3.58 seconds.

d. Summary of Parameter Estimation Results

Sections a. through c. presented the parameter estimation results achieved through each of the four approaches discussed. These include the observable Markov model and Gilbert model implemented by Schodorf as well as the continuous transmission functions implemented as part of this research (i.e., Gaussian and Hermite Interpolant). The next step is to compare the relative performance of the different approaches and determine which approach is the most accurate. For a Markovian process, the transition probabilities are directly related the mean state dwell intervals:

$$E[t_{dwell}] = \frac{1}{a_{ij}} = \frac{1}{1 - a_{ii}}. \tag{2.7}$$

Based on the dwell interval statistics presented in Tables 4 through 6, we can estimate the transition probabilities as shown in Table 15. It is important to note that these transition probabilities assume a time step of 0.1 seconds and a spatial distance step of 0.25 meters.

Table 15. Direct estimation of transition probabilities.

	a_{11}(connection)	a_{00} (fade)
Original data	0.9814	0.9405
Smoothed data	0.9874	0.9590
Spatial data	0.9931	0.9844

To compare the different continuous transmission HMMs with Schodorf's attempts, we need to analyze the results involving spatial data because that is the data Schodorf used. The parameter estimation results of the spatial data are summarized in Table 16. The best match is achieved with the mixed Gaussian transmission functions, indicated by the red ovals. The mixed Gaussian HMM provides a mean connection of 38.5 meters and a mean fade of 21.5 meters. These are both slightly longer than experienced by the empirical testing. The second best match is provided by the single Gaussian transmission functions; but the single Gaussian approach produces mean dwell

intervals that are slightly smaller than found in testing. The observable Markov model and Gilbert model do not provide nearly as good of a match with respect to producing mean dwell intervals.

Table 16. Comparison of parameter estimation results for spatial data

	a_{11}(connection)	a_{00} (fade)
Direct estimate (from Table 15)	0.9931	0.9844
Schodorf Markov (from Table 10)	0.9726	0.9465
Schodorf Gilbert (from Table 10)	0.9959	0.9264
Single Gaussian (from Table 11)	0.9923	0.9876
Gaussian mixture (from Table 13)	0.9935	0.9884

Table 17 provides a condensed comparison of the BWA results for the original and smoothed (temporal) data sets. Results are repeated from Tables 11, 13 and 14. In almost cases the parameter estimation has overestimated the state durations, especially with respect to the fade state and the original data set. The hermite interpolant approach consistently produces the largest estimates of transition probability regardless of state or dataset. These large transition probability estimates provide abnormally long fade and connection dwell intervals that are not reflected in the test data. Again, the two Gaussian-based approaches are closer to matching expectations and in some cases produce mean connection dwell intervals within a few seconds of the test data results.

Table 17. Comparison of parameter estimation results for temporal data

Original data:	a_{11} (connection)	a_{00} (fade)
Direct estimate	0.9814	0.9405
Single Gaussian	0.9820	0.9630
Mixed Gaussian	0.9867	0.9657
Hermite interpolant	0.9906	0.9713
Smoothed data:		
Direct estimate	0.9874	0.9590
Single Gaussian	0.9838	0.9670
Mixed Gaussian	0.9881	0.9677
Hermite interpolant	0.9910	0.9721

D. CHAPTER II SUMMARY

In this chapter, we achieved the first of the two research goals established in Chapter I, that of developing an improved urban COTM channel model. The mixed-Gaussian continuous transmission hidden Markov model is superior to prior attempts in two ways. The first is the increased accuracy in matching dwell interval statistics. The second improvement over previous models is the model's ability to produce a time sequence of signal strength that accurately matches the empirical training data. This is a direct benefit of the model being a continuous transmission HMM. This additional capability allows future model improvement to replace the visibility vector with a signal strength vector that includes other link-changing processes like weather or scintillation. The next chapter introduces the mitigation strategy that allows us to meet the remaining research goal of improving network performance in the urban environment.

III. INTRODUCTION: URBAN COTM AND AD-HOC PACKET MODELS

A solution for minimizing the impact of long outages experienced by the urban mobile-to-satellite link as discussed in Chapter II is presented in this chapter. The solution is to modify the packet routing protocol(s) to improve overall network performance either by reducing delay or increasing throughput. A discussion of several distinct but related models that were used to explore this solution path is included in this chapter. These models are focused on implementation of the network layer protocol as defined in the Open Systems Interconnection (OSI) reference model [48]. They also include some limited modeling of the Data Link and Physical layers on which the Network layer is built. The Physical layer is modeled as either a directional wireless link between a mobile radio and a geostationary satellite or a mobile radio to mobile radio wireless link experiencing slow, flat Rayleigh fading. The Data Link layer is only modeled in the mobile-to-mobile link with a convolutional code for forward error correction (FEC). The Data Link layer of the mobile-to-satellite link is assumed to include a combination of an error detection and correction algorithm and block interleaving that together provide a nearly error-free data stream to the network layer when the link is not experiencing fade due to building blockage.

Model 1 is presented as the control case, or baseline, for comparison. It simulates the performance of two independent packet routing protocols: a local ad-hoc network consisting of n mobiles and access to the larger GIG that is represented as a mobile-to-satellite link. It is assumed that the satellite is a central node in the GIG and transfers packets to their distant destination. In Model 1, the two networks are not integrated. When a mobile generates a packet, the mobile transmits the packet on the local ad-hoc network if the destination is local or transmits the packet on the SATCOM link if the destination is not local. Similarly, when the satellite node has a packet destined for one of the mobiles, the satellite attempts to transmit the packet directly to that mobile. If the mobile is experiencing a deep fade associated with physical blockage, then the packet is

lost and must be retransmitted at some later time. In this model, the satellite does not use any side information regarding the mobile's signal strength or lack thereof.

Model 1A is a slight variation of Model 1. This simulation explores possible performance improvements by modifying the mobile-to-satellite routing protocol to include a "wait-state" when the mobile appears to be blocked. The satellite estimates the link state (blocked or clear) using available side information on uplink carrier strength. The goal of this modification is to reduce the number of packets that the satellite sends to a mobile experiencing a long blockage. This in turn is expected to reduce the number of packets that have to be resent and improve overall throughput. In Model 1A, the local ad-hoc network is still separate and independent.

Model 2 implements the "Neighbor protocol" for comparison with Models 1 and 1A. The Neighbor protocol modifies the local ad-hoc network to interact with the SATCOM link in a more integrated fashion. The primary modification is to allow neighboring mobiles to act as surrogates for a blocked mobile and allow the network to circumvent long duration link outages. This modification essentially allows some packet routing paths to include both the local ad-hoc network and the satellite-node. The specific implementation represented in Model 2 is constrained to a single transfer within the ad-hoc network. This limitation is included for simulation simplicity. If the network performance shows improvements, the concept can be expanded to allow more complicated routing paths and more complicated routing protocols. The disadvantages of this approach are expected to be increased system complexity and increased traffic on the local ad-hoc network.

All three of these network models include internal Markov models for simulation of the urban deep fade phenomenon and simulation of the mobile's horizontal motion relative to its neighbors. The first of these Markovian phenomena is central to the performance of the SATCOM link. The second is central to the performance of the mobile ad-hoc network. An introduction to Markov models can be found in Chapter I. Chapter II discusses the empirical test data and Baum-Welch algorithm used to estimate the appropriate model parameters for the signal fade model. A third use of Markov

chains is to simulate the packet generation process. As discussed in [27], packet generation can be accurately represented using discrete Markov models. This portion of the simulation is discussed more in Section A.3.

Universal Assumptions

The Model 1 and Model 2 simulations are based on the following simplifying assumptions:

1) Single satellite: These models assume that each mobile has established a link with a single satellite for the entire duration of its TDMA BW assignment. Because the mobiles are using a directional antenna for the satellite link, it would be relatively time consuming to reposition the antenna and establish a new link (which would also mean a new TDMA slot assignment). The time necessary to reposition the antenna and establish a new connection is expected to be substantially longer than the average fade duration. Tables 4 and 5 show average fade durations of less than 3 seconds for the Boston data. It would be difficult to build a system that can frequently slew, settle and establish a connection in less than 3 seconds. The model does not necessarily assume that each mobile is communicating with the same satellite, but the models assume a strong signal-to-noise ratio (+10 dB) for all satellite links. See assumption 2 for additional details.

2) These models assume that the mobile-to-satellite link has an extremely strong signal with enough margin to make bit errors infrequent relative to the capability of any forward error correction. This assumption is applied equally to all three Models to keep performance comparisons appropriate. The purpose of this assumption is to focus on the deep fades and complete signal loss associated with moving through the urban environment. This assumption is reasonable for existing non-adaptive satellite links which are designed with enough margin to close the link even during rainy weather. As mentioned in Chapter I, the TCA may be implementing adaptive link control and may not operate with 10+ dB of link margin. While this assumption simplifies the current model, future simulation upgrades could easily include a computation of SATCOM link bit errors for comparison against error correction performance.

3) The corollary to assumption 2 is that the receiver experiences complete signal loss or considerably more bit errors than the FEC can handle when the SATCOM link experiences a deep fade. All packets (partial or complete) that are impacted by a deep fade are assumed to be lost and require retransmission. Like assumption 2, this assumption is made for simplicity and focuses attention on deep urban fades. When combined, assumptions 2 and 3 allow us to model the urban fade phenomenon as a simple two-state discrete Markov model. To add additional detail to the simulation,

we can replace the two-state discrete Markov model with a 2+ state model that has a probability distribution function associated with each state. This idea is also discussed in Chapter I [5], and [24].

4) The models also assume that both networks have the same number of symbols per hop, symbols per packet and hops per packet, where a hop is equivalent to a TDMA time slot. These assumptions are not necessarily required of the simulation or any real-world implementation. The models have been built with these assumptions for simplicity. They allow a reduction in total variable count and allow more code reuse between the two network simulations. These assumptions are also critical to implementing assumption 5.

5) While in a real implementation the two independent networks could easily run asynchronously, the simulation treats them as synchronous and models the satellite bit rate as an integer multiple of the ad-hoc bit rate. These assumptions are solely for the purpose of simplifying the simulation in both complexity and run time. By making this assumption, the simulation can operate on a single discrete time counter set to the shorter satellite hop duration. When the time counter increments to the appropriate integer multiple, the ad-hoc network transitions to a new hop period as well.

6) The model currently assumes each node (mobile or satellite) can only generate one packet per hop period. The purpose of this assumption is to provide unique identification of a packet by its origin, packet type and generation time. Overall, this assumption constrains each hop period to a maximum number of new packets equal to the number of mobiles plus one for the satellite. The generation of new packets at each node is modeled as independent processes. The packet generation methodology is discussed in more detail in Section B.1.d. In each case the packet destination is selected randomly with uniform distribution.

A. MODEL 1: INTRODUCTION AND DISCUSSION

Model 1 represents the comparison baseline. In Model 1, two parallel networks are operating. The first is the mobiles-to-satellite network, with the satellite acting as a hub. The second is the local area ad-hoc network between neighboring mobile nodes. In this section, we will introduce both networks and general packet management processes (generation, tracking and control).

1. Mobile-to-Satellite / COTM Link

The heart of this investigation is the mobile-to-satellite link because that is the communication channel that experiences long outages in the urban environment. Figure 33 provides a highly simplified flow diagram of the basic COTM link as implemented in the Model 1 simulation. The COTM channel simulation is introduced in this section and modifications are presented in Sections B (Model 1A) and C (Model 2).

Modern communications satellites like TCS are expected to become a central part of the Global Information Grid (GIG). Like many organizations and communities, the military is pursuing a future communications architecture that supports real-time battlefield awareness at the lowest levels (e.g., individual vehicle). Another goal is to get more timely and accurate information to the individual soldier. The solution to both of these goals is a larger, more complicated communications architecture with a substantial increase in the number of simultaneous users and an increase in the number of occasional users that come and go.

One approach to solving the simultaneous user problem is to simply load the satellite up with one transceiver for each user. This approach would be inefficient, prohibitively expensive on the satellite and would require a large amount of frequency spectrum [49]. So under the constraints of limited spectrum bandwidth and a finite number of receivers, the solution needs to be more efficient. This problem is similar to the cellular telecommunications problem with large numbers of cellular phones passing in and out of geographic cells. Like cellular systems, we need to look at more modern communication approaches like code-division multiplexing algorithms (CDMA) or time-division multiplexing algorithms (TDMA). This protocol and model use TDMA which is discussed more below.

Another impact of simultaneous and occasional users is the use of packets and packet protocols. In this situation, the communications satellite becomes a packet server within a Local or Wide Area Network (LAN/WAN). This approach is highly flexible and handles traffic that comes in bursts. Packet protocols allow the system to handle data

transfers in smaller pieces. There is a rich field of research into packet protocols which can be leveraged by the satellite community. Packet routing protocols are also powerful at working around local area jamming, network outages or congestion.

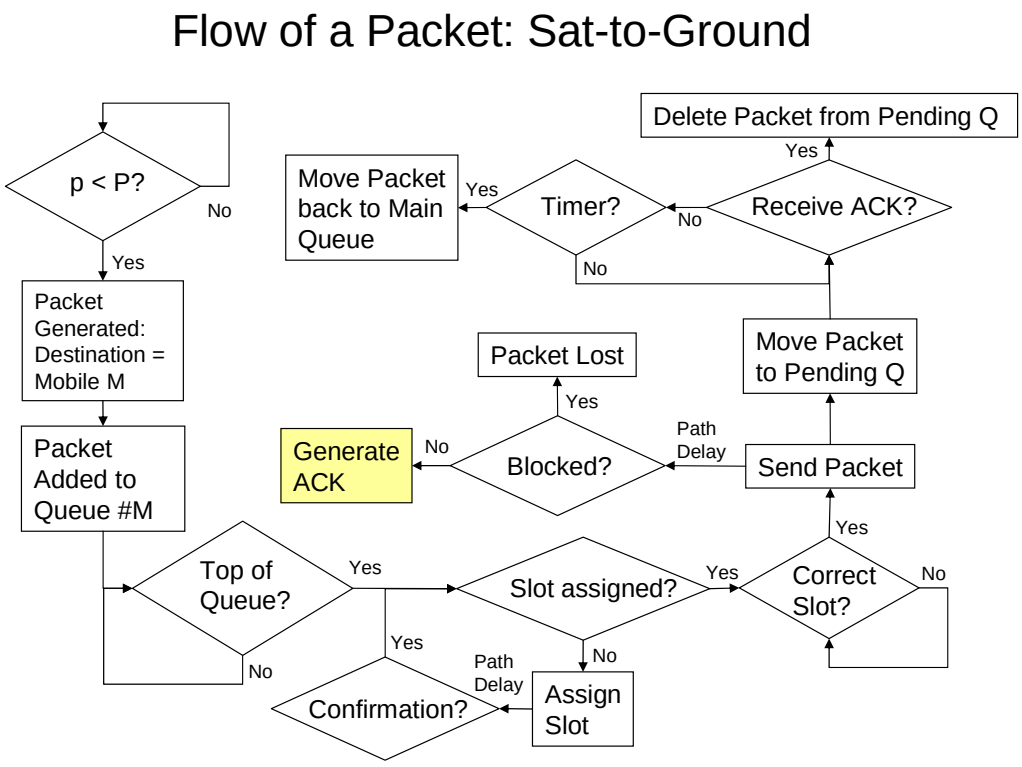

Figure 33. Flow chart of COTM link processes.

While the DoD's TCA effort is attempting to leverage technology from the commercial internet, it must also plan to incorporate military-unique features to provide communications and transmission security [1, 3]. The TCA has important requirements to provide anti-jam and low-probability-of-intercept (LPI) communications. Frequency-hop spread spectrum (FHSS) is a common but powerful method of achieving this goal. FHSS is already a common technology in existing military communications systems. This is likely to continue in the future. Because it is not a central area of investigation

and does not directly impact network performance, the simulation does not currently model the specifics of the frequency-hop (FH) pattern but one is assumed. FH communication systems are well understood and are discussed in references [50, 51]. Fortunately, the model results are independent of whether the communications link is based on FHSS. For the purposes of this dissertation, the term "hop" is in reference to the assumed FHSS implementation and is synonymous with a TDMA time slot.

a. Time Division Multiplexing Algorithm (TDMA)

The mobile-to-satellite communications model is based on the demand-assignment multiple access (DAMA) version of TDMA [49]. By implementing DAMA-TDMA, capacity assignments are not fixed and are changed as needed to respond to changes in customer activity. A new mobile that wants to be added to the network must request a time slot. The model assumes that this activity occurs on a separate control channel set aside for network administration traffic. When the satellite receives a request for a new time slot, it must notify all of the mobiles of the new order. This is also done on the administrative channel. This aspect of the communications architecture is not modeled.

For maximum efficiency, the duration of an individual time slot should be an integer fraction of the packet length. This makes the total time to transmit a packet $\left[(n-1)m+1\right](t_{timeslot})$ where m is the number of users with assigned time slots, n is the number of time slots required to send a whole packet and $t_{timeslot}$ is the duration of a single TDMA time slot. If we assume that the packet size is fixed, $t_{timeslot}$ is also a function of n: $t_{timeslot} = packet\ size \times t_s / n$ where t_s is the symbol duration. For the results presented in Chapter IV, n = 8, and a TDMA time slot is 1/8 of the total packet length. The model also assumes 1024 symbols per packet and 1 μs symbols. Using these simulation assumptions, we can update the total packet transmit time:

$$\left[(n-1)m+1\right](t_{timeslot}) = (7m+1)(128\mu s). \tag{3.1}$$

As the ratio of packet length to TDMA time slot duration increases, the total time to transmit a packet grows due to the multiplicative factor m. This, in turn, increases the probability that a packet is interrupted by a blockage and decreases the protocol efficiency.

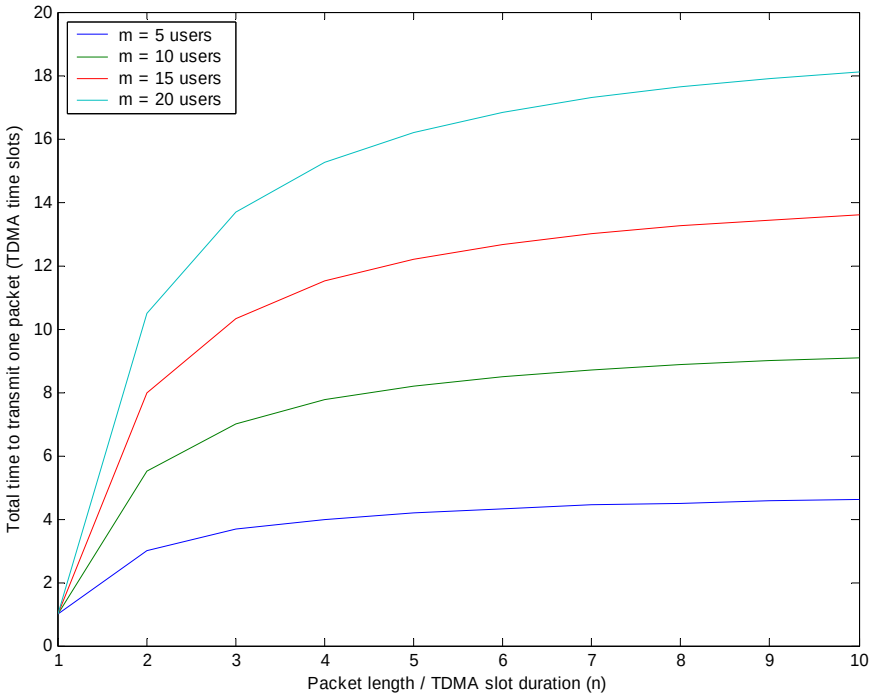

Figure 34. Total packet transmit time vs. TDMA slot size.

Figure 34 shows this relationship for a range of active users in the network m. It is also important to note that hop duration should also be integrally tied to TDMA time slot duration and packet length as an integer multiple or integer fraction. Because the probability of the mobile node transitioning from the connection state to the fade state is the same for every time step, the probability of successfully transmitting a downlink packet is

$$\Pr(successful\_packet\_Tx) = a_{11}^{\frac{(n-1)m+1}{n}} \tag{3.2}$$

Using the connection transition probability a_{11} from the empirical Boston data, we can see the impact to probability of successful packet transmission in Figure 35.

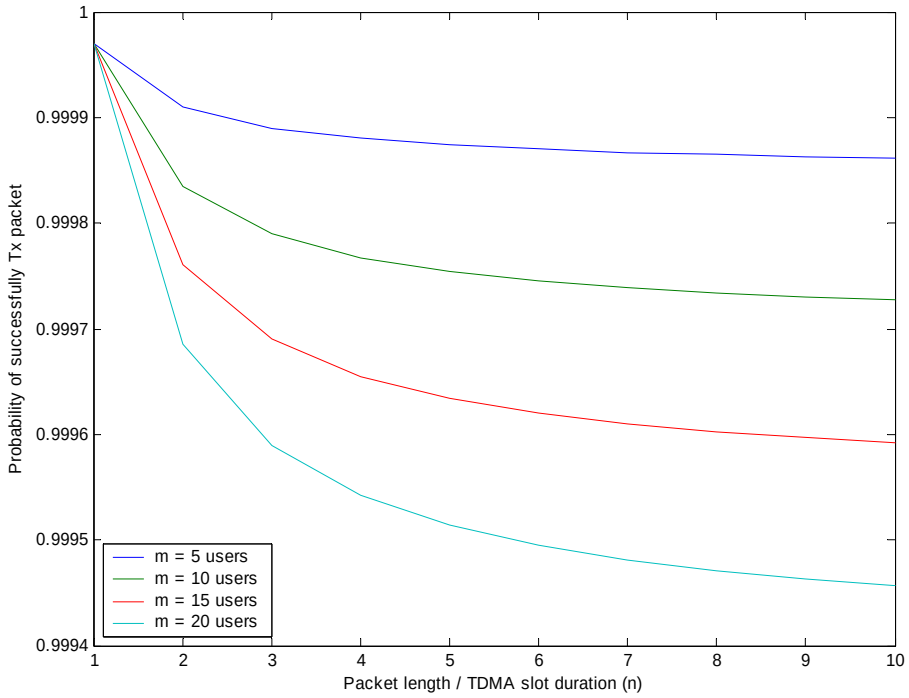

Figure 35. TDMA impact to probability of successful packet transmittal.

b. Path Delay

One of the more unique facets of satellite communications is the path delay. It is a direct result of the large distances between the earth-bound customer and the geosynchronous satellite. The specific path delay between a user and the satellite is dependent on the actual distance. The minimum path delay occurs when the user is directly below the satellite, known as the zenith (Draw a straight line between the center of the Earth and the satellite and the mobile is along the same line.). In this situation, the path delay is

$$\frac{r}{c} \approx \frac{35,786}{299,792.5} = 0.119369 \text{ sec}, \tag{3.3}$$

where r = separation distance = geosynchronous altitude = 35,786 km,
c = speed of light (vacuum) = 299,792,458 m / s.

The average path delay is generally higher, mostly due to *r* being larger, but also due to the speed of light in atmosphere being slightly smaller than that of vacuum. As the mobile gets farther from the sub-satellite point, the separation distance grows to a maximum of 41,678.936 km. This maximum assumes that the satellite is on the mobile's local horizon (elevation = 0 degrees). This maximum can be slightly higher if the mobile wanted to establish a connection when the satellite was below its local horizon (elevation < 0 deg). In this case,

$$\frac{r}{c} \approx \frac{41,678.9}{299,792.5} = 0.139026 \text{ sec.} \tag{3.4}$$

The simulations assume a path delay of 0.125 seconds, which represents the satellite being halfway between the mobile's zenith and horizon (satellite is 44.13 degrees above mobile's local horizon). Due to the semi-spherical nature of the Earth, the path delay increases slowly as the satellite moves away from the mobile's zenith and increases rapidly as the satellite approaches the mobile's local horizon. For a data rate of 1 Mbps, this path delay is equivalent to 125,000 symbol durations.

Terrestrial path delays are traditionally orders of magnitude smaller because the distances are orders of magnitude smaller. This large path delay has a significant influence on any SATCOM protocol performance. As the number of interactions or queries between satellite and mobile increases, so does the protocol inefficiency. This problem is one of the central rationales for not using an asynchronous, ad-hoc protocol like that recommended for the mobile-to-mobile link. Instead, the mobile-to-satellite protocol is based on assigned TDMA time slots that require one mobile-to-satellite interaction cycle to establish but then allows easy packet transfers without any additional overhead like the *request-to-send* (RTS) message exchanges found in the ad-hoc protocol discussed in the next section. This approach does require reassignment of time slots whenever a user joins or leaves the network. If this is a frequent event with respect to path delay durations, then TDMA may not be the best approach. Another important consideration when using TDMA is the percentage of empty time slots for the average user. If packet transfers are relatively infrequent, a substantial portion of the data bandwidth is being wasted.

One skewed impact of path delay is the impact on accurate side information regarding a node's visibility state (see next section). The mobile node and the urban structure that is blocking its line-of-sight are both terrestrial and relatively close physically. The mobile transceiver can have nearly real-time knowledge of its own connection state by simply monitoring the signal strength of the downlink, but the satellite is as far from the physical fade source as possible. Any change in signal strength will be available to the satellite after incurring the full path delay. This limits the usefulness of the side information available to the satellite. To make matters worse, any modification to the satellite's behavior will not be available at the mobile for another full duration of the path delay. This impact plays a significant role in the performance of Model 1A and is discussed more in Section B. Path delay also impacts the difficulty of predicting COTM channel state based on the satellite's knowledge of signal strength. This topic is discussed more in Appendix A.

c. Visibility (Clear vs. Blocked)

A central goal of this investigation is to compare performance of packet protocols in the urban environment with frequent and extended mobile blockages. As shown in Chapter II, Markov models are a good approach for analytically modeling link fade due to blockage. The Markov model implemented in this larger packet protocol model is not hidden and has only two states: clear and blocked. The blocked state is defined as signal strength too low for the mobile receiver to detect. The clear state is defined as high signal strength with very low bit error rate that is easily corrected by a forward error correction scheme. This is an ideal assumption that simplifies the COTM link model by not dropping packets due to individual symbol errors. It is assumed that the building blockage would be the dominant cause of lost or damaged packets. A key recommendation for future simulation improvement is to replace this simple, observable Markov model with the mixed-Gaussian continuous output HMM derived in Chapter II. The addition of the more complicated visibility model would produce estimates of signal strength as a function of time and allow incorporation of secondary link impacts like scintillation or weather.

For the visibility model itself, the behavior and parameters of the Markov model are assumed to be known. As discussed in Chapter II, appropriate values for transition probability were computed using a Baum-Welch algorithm and empirical data from Boston, MA ($a_{00} \sim 0.95$, $a_{11} \sim 0.975$). Because the simulation operates at a much finer time step than the empirical data, the transition probabilities must be scaled. The equivalent transition probabilities at 0.000128 seconds per time step are ($a_{00} \sim 0.999936$ and $a_{11} \sim 0.99997$). These transition probabilities produce mean connection and fade durations of 4.2 and 2.0 seconds, respectively.

Figure 36 shows a 300,000 sample example of a visibility input vector for one mobile. The *x*-axis has been converted to seconds. The 38 second example includes ten state transitions.

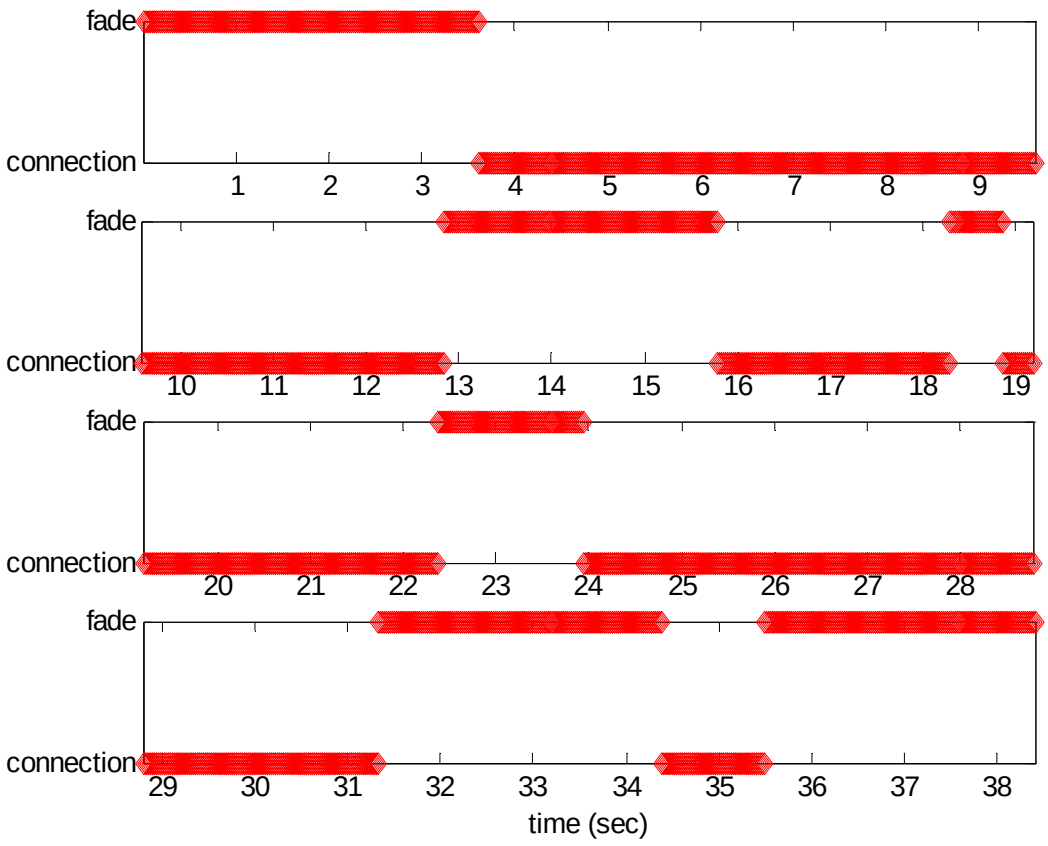

Figure 36. Example visibility vector.

2. Mobile-to-Mobile (Ad-Hoc) Link

The primary focus of this simulation and the surrounding research is to circumvent or overcome the long term deep signal fade that occurs to a mobile node in the urban environment. A key supporting facet of the investigation is the possibility of the individual mobile being part of a terrestrial ad-hoc network. In the military version of this situation, the mobile ad-hoc network is expected to be a platoon of HMMWVs, perhaps on a combat or peacekeeping mission. A mobile ad-hoc network is defined as:

> *a self-configuring network of mobile routers(and associated hosts) connected by wireless links – the union of which form an arbitrary topology. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet.* [52]

This simulation does model a terrestrial mobile ad-hoc network as part of a larger network which includes the satellite communications infrastructure. It is important primarily because the Neighbor protocol implemented in Model 2 takes advantage of these additional resources. But Models 1 and 1A do not assume that the mobile ad-hoc network is integrated with the larger SATCOM network. In Models 1 and 1a, the mobile ad-hoc network is operating independently, and performance results are useful only as a baseline for understanding the impacts of integrating the ad-hoc network with the SATCOM network. A simplified flow-chart overview of the network processes is provided in Figures 37 and 38. The specific mobile ad-hoc packet routing protocol implemented in these simulations is loosely based on the work of Pursley et al. [53, 54, 55, 56, 57, 58, 59]. This particular implementation was selected due to the strong connections to existing and planned military communications systems (e.g., SINCGARS). The selected implementation also matches many aspects of the COTM link which eases network integration. These similarities are discussed more in the next section. This particular group of authors, which primarily includes Pursley, Russell, and Wysocarski, have explored several variations of the *least resistance routing* (LRR) forwarding/routing protocol with increasing complexity. This research is not focused on

the ad-hoc portion of the network, so a fairly simple version has been selected for implementation. In Chapter V, the recommendations for future work include the possibility of upgrading the ad-hoc network to reflect a more complicated approach that could include adaptive transmission, packet type prioritization or routing metric improvements. In addition to the technical papers listed above, background information can also be found in [60] and [61].

Mobile ad-hoc networks are highly dynamic networks in any case due to mobility of the nodes, variability in the propagation conditions and variability in the interference. A military version of the mobile ad-hoc network is even more dynamic based on the possible destruction of a node due to combat activity and directed jamming from hostile forces [56]. The wireless ad-hoc packet network is a valuable communication architecture for the military because they are expected to operate in situations where fixed infrastructure is not available. The ad-hoc packet protocol presented here and in [53-57] shares many aspects with maturing commercial solutions, but there are significant differences. One significant difference is the requirement to operate while experiencing directed hostile jamming. Another important difference are the heightened security requirements to minimize the probability of intercept.

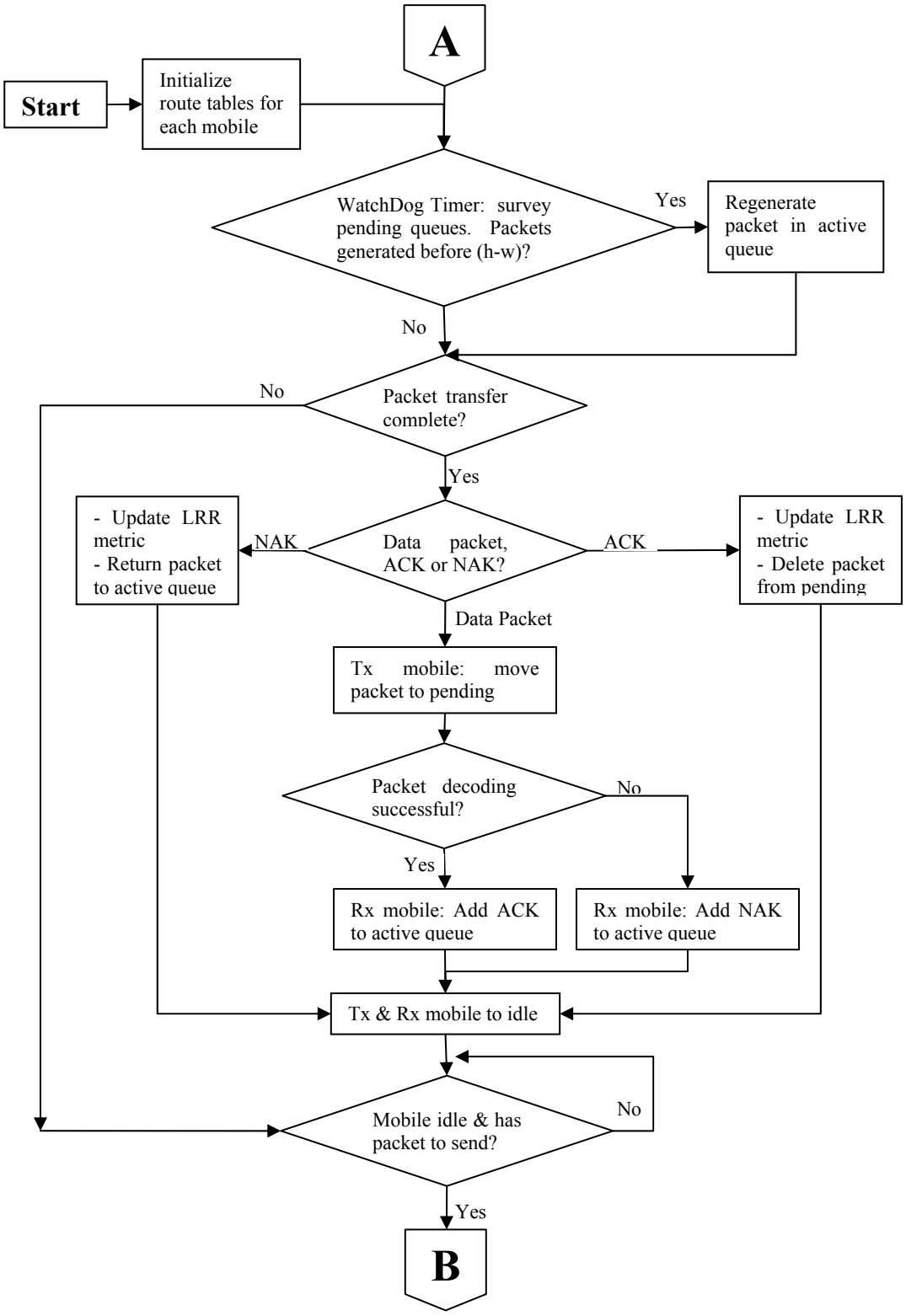

Figure 37. Overview of mobile ad-hoc simulation.

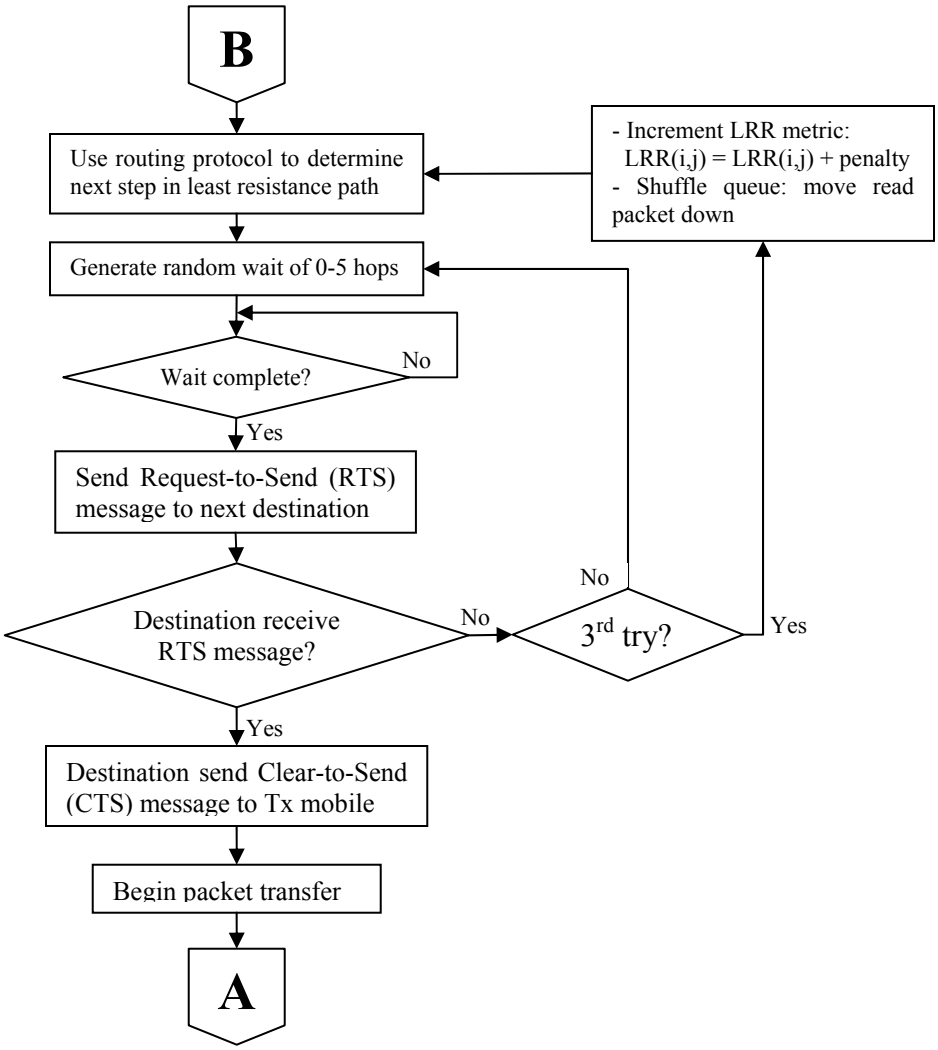

Figure 38. Overview of mobile ad-hoc simulation (cont.).

a. Mobile Ad-Hoc Network Design: Background and Assumptions

In [53] through [57], Pursley, Russell, Wysocarski and others investigate several different implementations of proactive, table-driven, adaptive packet routing protocols as applied to a tactical, wireless packet-radio network. These investigations usually include network modeling and performance results for protocol comparison. The technical investigations primarily focus on mobile radio nodes employing time-slotted FHSS signaling in a packet-based network. Another common aspect of this body of work is the use of FEC coding, with a specific focus on Reed-Solomon block coding. As

discussed in the previous section, the SATCOM COTM link is also assumed to be time-slotted (TDMA), FHSS and packet-based with FEC. These similarities allow the two protocols to work more efficiently together.

As described in [53], "each receiver has a set of hopping patterns, one for each hop period in which a [packet transfer] may be initiated. This is called a 'receiver-directed hopping pattern' [53, 54, 56]. A transmission to a given receiver must use the correct hopping pattern for the receiver and the dwell interval in which the transmission begins. By transmitting the packet using this hopping pattern, it is extremely unlikely that any radio other than the intended recipient will succeed in dehopping the packet." Please note that a *dwell interval* is defined as the duration of a hop which is different from a Markov process' dwell time (e.g., interarrival time).

Another important assumption for this network design is that the mobile radio systems have a half-duplex transmission capability. (The satellite is assumed to be full-duplex with separate transmitter and receiver of different frequencies). This is considered a reasonable but conservative assumption with respect to analyzing the performance improvements of the Neighbor protocol. Physical space is limited in military vehicles like the HMMWV, and minimizing the size of the unit is important. The half-duplex mobile-to-mobile transceiver is in addition to the full duplex system that is communicating with the satellite. It will be difficult enough to accommodate that much hardware on each and every vehicle. The half-duplex transceiver assumed for the mobile ad-hoc network is considered a minimum capability necessary to support a minimally capable network. If the mobiles are able to carry full duplex systems, then the performance of the Neighbor protocol will improve.

There are three situations that prevent a packet from being received: 1) the intended recipient is already busy, 2) two or more mobiles attempt to start a transmission during the same hop period, to the same recipient (at the same frequency), and 3) there are too many errors in the transmission and the receiver is not able to decode the packet.

Another key aspect of the referenced routing protocols is the use of active acknowledgements [53] with a selective reject version of automatic repeat request

(ARQ). This is the same form of ARQ implemented in the COTM channel protocol. Using the same ARQ method in the COTM channel and the ad-hoc network helps ease their integration in Model 2. The benefits and performance of selective-reject ARQ are discussed further in Section A.3.

A packet is defined to consist of L codewords with each codeword containing n symbols for an (n, k) FEC block code.

While different versions of the protocol focus on using different side information, the Pursley protocols all use side information to estimate the optimal packet routing through the network. Variations include an embedded string of test symbols in each hop period. The receivers analyze the test symbols for errors and use the number of errors to estimate link quality. Another idea is to estimate the number of bit errors in the packet content through application of a forward error correction code.

b. Packet Routing: Forwarding Protocol

As mentioned earlier, mobile ad-hoc networks in general, and military versions specifically, need to provide robust communications in a fast changing environment. The network can expect to add or subtract nodes and experience time-varying link performance due to internal or external interference and node motion. This situation is the impetus for focusing on a packet routing forwarding protocol instead of just a routing protocol. The difference between *forwarding* protocols and *routing* protocols is one of perspective and speed. Forwarding protocols are narrowly focused on a small piece of the network topology and the short period of time. They determine a path for forwarding a packet one node closer to its destination based on the current, local conditions [54]. The forwarding protocol is focused on a single link between one node and another and is able to estimate the quality of this link based on the most recent packet or acknowledgement from that node. This simplicity allows the forwarding protocol to react faster to changes in the network. On the other hand, routing protocols determine route metrics and traffic patterns between every pair of nodes in the network. The process is clearly more complicated and requires additional time to generate and

propagate to all nodes. The routing protocol is better suited to modifying the network for long-term or global changes in the network, but the forwarding protocol is better suited to modifying a traffic pattern for short-term or local changes. The LRR protocol discussed in this section is a forwarding protocol that can also act as an adaptive routing protocol by sharing resistance metrics amongst nodes. This can be accomplished as part of the frame header of a standard data packet or as separate traffic control and maintenance packets [54].

A central concept to this forwarding/routing protocol is the use of side information to calculate the LRR metric [54, 55, 56]. The metric is intended to provide a quantitative comparison of the probability of delivering the packet within the specified requirements for delay or bit error count. One of the most basic versions of the LRR metric is the errors and erasures (EE) metric [54]:

$$LRR_{EE}(A,B) = e_1 + 2e_2, \tag{3.5}$$

where e_1 = number of erasures (hop periods erased due to error count > threshold), and
e_2 = number of symbol errors per block.

This basic metric computation relies on side information consisting of the number of errors and erasures that occur at the receiver. These can be accurately computed by sending a string of test symbols at the start and finish of the hop period. These test symbols are in addition to the packet content. If there are too many errors in the string of decoded test symbols, then the protocol will erase the entire hop period. This is counted as one *erasure*. If the number of errors in the test string is below the threshold, then the information can be used by the decoder to improve performance and estimate the number of errors in the hop period. This data can, in turn, be sent back to the transmitter in the acknowledgement for update of the LRR metrics.

Reference [54] also proposes a second, slightly more complex metric that the authors refer to as the errors, erasures, and acknowledgements (EEA) metric. This metric is the same as the EE metric except when no acknowledgement is received. In this case, the transmit node increases the LRR metric by adding a constant c_a. The author

suggests a value $c_a \approx 1$. For the basic EE metric, if the node does not receive an acknowledgement, it does not update its LRR metric and may continue to incorrectly select the same recipient.

In [55], packet delay requirements drive the addition of a new term W(B), which is referred to as the *node-dependent resistance*. The node-dependent resistance estimates the possible delay the packet will experience while stopping at that node before being forwarded. It may be based on side information such as the number of packets in that node's buffer, the amount of other traffic near that node or the delay experienced in getting the most recent acknowledgement back from that node. This last approach requires the protocol to not prioritize acknowledgements so that they wait in the queue like any other packet. The initial EE or EEA metric becomes the other half of this new metric and is considered the *link-dependent resistance*, which estimates the quality of the link between nodes A and B. It attempts to account for multipath fading, path loss, interference, etc. These two resistance terms combine to form the updated routing metric:

$$LRR(A,B) = \alpha I(A,B) + \beta W(B). \tag{3.6}$$

Each of the two resistance terms has a weighting coefficient (α, β) that allows the metric to be tailored to the relative importance of specific network requirements.

(1) Asynchronous traffic. As mentioned earlier, the mobile ad-hoc protocol is designed for a half-duplex system. This constraint has a significant impact on protocol design. Because each node must transmit and receive but can do only one activity at a time, the mobiles must take turns or risk frequent multiple-access interference problems. If a mobile has a packet to transmit, it must first establish permission to do so, which we will refer to as a connection. This mobile-to-mobile connection is temporary and only applies to the transfer of a single packet. The connection is accomplished by the transmit node sending a short request-to-send (RTS) message. If the receive mobile is already busy (e.g., transmitting or receiving), then the RTS message is ignored and the transmitting mobile waits a random number of time steps before trying again. If, instead, the receiving mobile is not busy and is able to decipher

the RTS message (relatively few errors), then it replies to the transmit mobile with a short clear-to-send (CTS) message. In the simulation, these messages are both implemented as one time-step (128 μs) in duration. After receiving a successful CTS message, the transmitting mobile begins immediate transmission of the actual data packet. When the data packet transfer is complete, the connection is complete and the receiving mobile generates an acknowledgement that waits its turn in the active queue. The connection "hand-shake" needs to be repeated for the acknowledgement at some later time. This process is different for networks that employ single-threading which is discussed below.

(2) Side Information, Each approach to computing the LRR metric relies on some form of side information. In some cases, the side information is assumed to be perfect (e.g., exact value, instantaneous knowledge) when in a real implementation this is not the case.

The basic link-dependent resistance term *I(A,B)* requires the number of errors and erasures that occur at the receiver. These can be estimated by inserting a string of test symbols at the start and finish of the hop period. If there are too many errors in the string of decoded test symbols, then the protocol erases the entire hop period. This is counted as one *erasure*. If the number of errors in the test string is below the threshold, then the information can be used by the decoder to improve performance and estimate the number of errors in the hop period. In this research, erasures are not modeled and the link-dependent resistance term is simplified to simply counting errors. This simplistic approach reduces some complexity of the simulation and the protocol. It requires an assumption that the FEC code can identify any number of errors, which is not true.

This approach does not represent an assumption of perfect side information. The most recent error count is only an short-term estimate of link quality. An assumption of perfect side information is counter-productive in this case because it would obviate the need for a routing protocol at all. The network would always know the best possible path between any two nodes.

(3) Single-threading. In the context of mobile ad-hoc networks, single-threading is a term applied to the concept of making the acknowledgement an immediate and integral response to a data packet transfer [53]. In single threading, a complete packet transaction consists of four immediately sequential steps: 1) Source: RTS message, 2) Destination: CTS reply, 3) Source: data packet, 4) Destination: acknowledgement. These are the same four steps completed without single-threading, but the acknowledgement is not an immediate response to the data packet transfer. Instead, the acknowledgement goes into the *first-in, first-out* (FIFO) queue and waits its turn like any other packet. The benefits of single-threading are an decrease in overall network latency and increase in throughput performance. The downside is that this approach removes a source of side information regarding node-dependent resistance. The simulation does not currently employ single-threading. It is a suggested area for further investigation.

(4) Implementation. Models 1 and 2 implement an LRR metric that resembles Equation (3.6) above with several minor alterations. While Pursley et al., base their link-dependent resistance term on a straight count of erasures and errors, we instead use bit error ratio (BER). These two approaches represent similar side information but have different scales. The Pursley et al., approach to computing $I(A,B)$, has an integer value from 0 to the number of hops in a packet (e_{max}). (If all of the hops are erased, then e_2 is zero.) By using BER for a single packet, the link-dependent resistance takes on a fractional value from 0 to 1. The same difference is true for the node-dependent resistance. A simple count of the number of packets in node B's queue provides an integer value from 0 to the buffer size. The simulation instead divides by the buffer size to produce a fractional value between 0 and 1. This starts the two resistance factors with equal weighting. In the Pursley approach, the same results/effects can be accomplished by incorporating the additional scaling factors into the weighting coefficients.

A second alteration is the inclusion of two additional terms or factors. One of these additional terms is the acknowledgement factor c_a mentioned

earlier when discussing the EEA metric. The second factor follows the same concept but instead increases the LRR metric when a node sends multiple RTS requests but does not receive a CTS reply. This term has not been proposed by Pursley et al., but we found it to be an important addition to the adaptive nature of the metric. Without this additional term, a mobile may repeatedly query a distant neighbor but receive no reply; the transmitting node would not make any changes to the routing table and continue to query the distant neighbor despite the lack of response.

The third alteration to the basic routing protocol is a steady time-based erosion of all LRR metrics by some small amount ε. The purpose of this additional modification is to force the protocol to occasionally reassess the quality of a network path. Without this alteration, we found that a path metric LRR(A,B) could have a large metric even though the two nodes are moving closer and link quality had seen a large improvement. Since that path has a large metric, it continues to be avoided by the routing protocol.

A fourth and final alteration involves averaging the routing metric updates with the previous estimates. This provides a smoothing to metric fluctuations that increases the period of protocol oscillations and minimizes erratic behavior [49]. This same approach was used by third generation ARPANET in 1987.

The simulation protocol treats all packets equally and does not attempt to differentiate between packet types with different performance requirements (e.g., latency vs. error count). This concept is proposed by Pursley et al., in [55], but is proposed as future work. The concept is discussed further in Chapter V.

Figures 39 and 40 show two examples of the LRR metric fluctuations as functions of time. Each figure represents the $n-1$ metrics that each mobile maintains with respect to each of its $n-1$ neighbors.

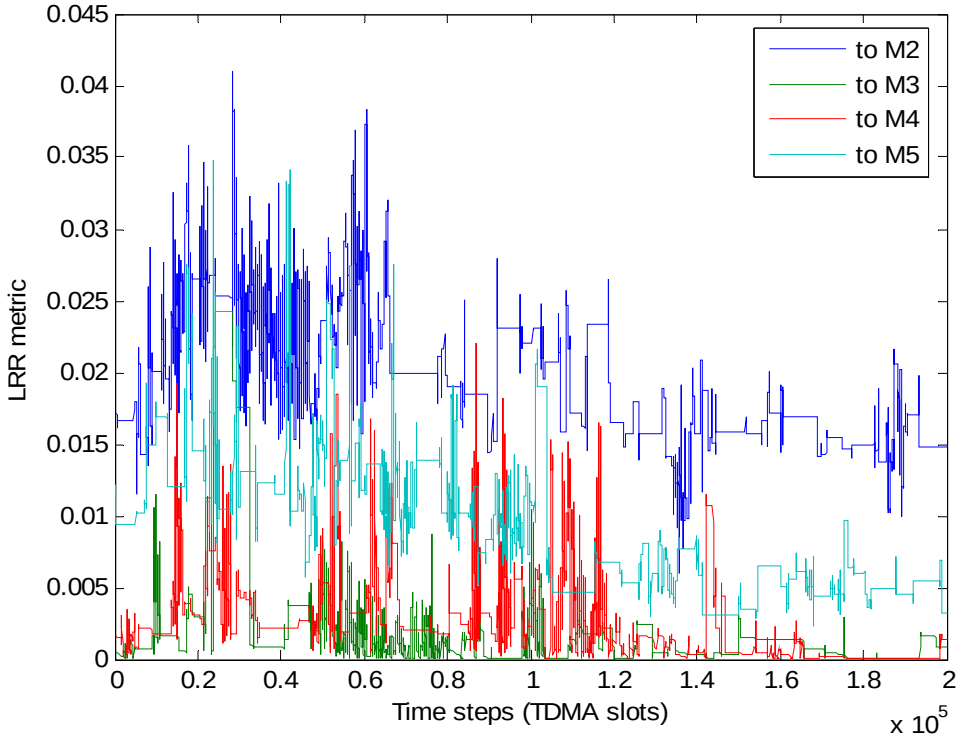

Figure 39. Mobile 1's LRR metric vs. time (Model 2, $r_p = 0.008$).

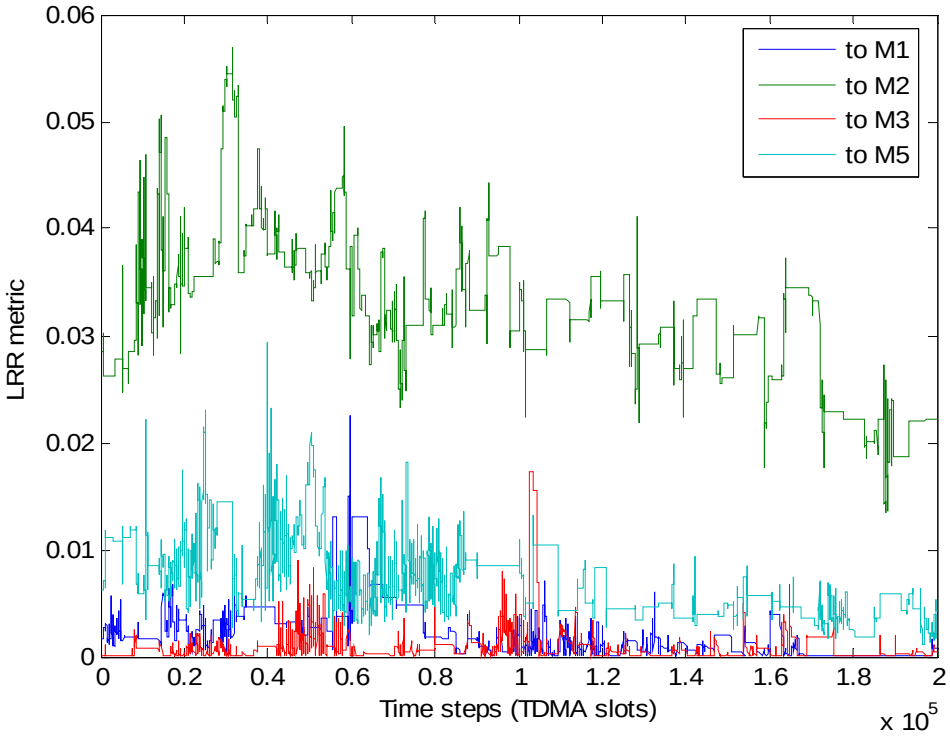

Figure 40. Mobile 4's LRR metric vs. time (Model 2, $r_p = 0.008$).

References [55], [56], [58], and [59] offer additional modifications, complexities and improvements to the LRR metric calculation. In Chapter V, we recommend these kinds of modifications to the ad-hoc network simulation for future work.

c. Other Traffic Management: Packet Shuffle

Packet shuffling is a traffic management support process associated specifically with the ad-hoc network but not directly involved with determining a packet's path through the network. Just as the name implies, packet shuffling involves the simple activity of moving packets within the queue. The simulation implements this process when a node makes several attempts to establish a connection with another node by sending a RTS message but fails to receive a reply. One possible reason for this type of problem is that the receive node is too far away and cannot successfully decipher the

RTS message. The LRR metric computation handles this problem with the addition of a constant discussed earlier. Another possible reason for this type of situation is that the receive node is already busy. In that case, the packet shuffle process helps by shifting the packet a few positions deeper into the queue and refocusing attention on the next packet in line. If the next packet has the same intended recipient, then nothing is lost or gained. If the next packet has a different path through the network, the new receive node may not be busy and the transmit node can continue to achieve positive progress in pushing data through the network. The overall goal is to improve network performance by minimizing the packet service time.

d. Vehicle Motion

To more accurately reflect reality, the model does include vehicle motion which is assumed to random and discrete in both time and space. The vehicles move within a limited Cartesian grid of 350 x 350 meters. The outer limits are set to keep the vehicles within reasonable range of another mobile node but far enough to make some links unmanageable and force multi-step packet transfers through the network. The discretization in time is by TDMA time slot which is set to 128 symbols × 1 microsecond per symbol = 128 microseconds. This discretization is so fine relative to urban vehicle speeds that it appears to be nearly analog. The discretization in distance is set to match reasonable vehicle motion in an urban environment. Assuming an average speed of six meters per second, or approximately 13 miles per hour, the distance per 128 microsecond time slot is 0.000768 meters (less than a millimeter). This divides the 350 x 350 meter box into an extremely fine grid of discrete locations.

The only purpose of the vehicle motion model is to more accurately reflect the real performance of the mobile ad-hoc links under time varying separation distance. The vehicle motion is not directly used in modeling the SATCOM link performance. The vehicle motion and the visibility models are not correlated. As an example, the vehicle motion model may have the vehicle stay in the same place for the next time slot, but the

visibility model may show a transition from clear to blocked. This type of correlation would add unnecessary complexity to the model without relevant impacts on the results.

Like the visibility phenomenon, vehicle motion is modeled as Markovian to capture the stochastic nature. The rationale for this selection is the basic idea that in any given 128 microsecond time slot, the vehicle tends to stay moving in the same direction. Changes in direction are a significant event and occur less often. This type of behavior is easily represented by a Markov model with the appropriate state transition probabilities (e.g., $a_{ii} >> a_{ij}$). The vehicle motion model computes two independent Markov models simultaneously, one for each Cartesian direction (x, y). Each of the two Markov motion models has three states: 1) move +1 step, 2) move −1 step, 3) stay in place.

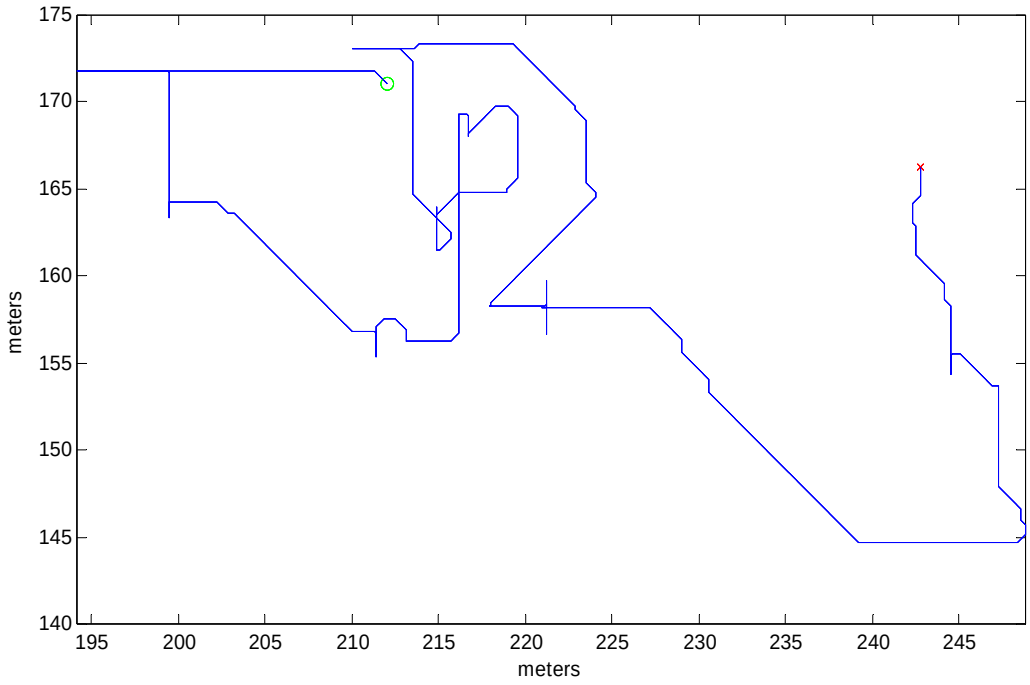

Figure 41. Example: 42 seconds of vehicle motion within 350x350 meter Cartesian grid. Green circle represents starting location, red "x" represents finish location.

When combined, the two Markov motion models result in a vehicle that moves in one of eight directions or stays in place. Whichever solution the model

produces, the vehicle tends to stay moving in that same direction. In the rare instance that one of the models changes states, it most likely makes a slight change in motion (45 degree dogleg in x velocity or y velocity only). An example of one mobile's motion is shown in Figure 41. This example reflects an a_{ii} of 0.99998, which provides a mean state duration of 50,000 time steps, or 6.4 seconds (assuming a time step of 128 μs).

e. Link Quality: Slow, Flat Fading

Unlike the COTM link, the ad-hoc link simulation does include the generation of individual symbol errors. The path distance between nodes is time-varying and a significant influence on link performance. And time-varying link quality is a driving factor in the performance of the ad-hoc (Pursley) forwarding protocol. One of the primary factors in computing a path through the network is link quality as estimated by counting bit errors corrected by the FEC on recent packets. Without time-varying link quality, a portion of the forwarding protocol becomes static.

In this urban environment and with omni-directional antennas, there is a significant amount of multipath signal energy. For the purposes of keeping the simulation simple, we assume slow, flat Rayleigh fading and M-ary phase-shift keying (MPSK) modulation. The probability of symbol error in this situation is

$$P_s \cong 1 - \sqrt{\frac{\overline{\gamma}_s \sin^2 \frac{\pi}{M}}{1 + \overline{\gamma}_s \sin^2 \frac{\pi}{M}}}, \tag{3.7}$$

where M = modulation order, and
$\overline{\gamma}_s$ = ratio of the average energy per symbol-to-noise power spectral density.

This relationship is also an important factor in sizing the vehicle motion grid discussed in the previous section. With the assumption that the FEC code can handle one error in eight symbols, we can explore the distance at which the ad-hoc links transition from mostly successful to mostly not successful. For the assumptions used in this simulation, that distance is approximately 339 meters. The maximum distance

between any two mobiles is 495 meters (opposing corners). Initial vehicle location is modeled as a uniform random variable. Since the motion process is Markovian with equal probability of moving in any given direction, instantaneous vehicle location is also modeled as a uniform random variable. This allows us to determine probabilities based on area. In the worst case that a mobile is in a corner, the probability of one neighbor being within 339 meters is

$$\frac{\text{Portion of area within reach}}{\text{Total area}} = \frac{\frac{1}{4}\left(\pi r^2\right)}{l^2} = \frac{\frac{1}{4}\left(\pi(339^2)\right)}{350^2} = 0.737. \qquad (3.8)$$

With n mobiles, the total probability of having at least one mobile within range is $\left[1-\left(1-0.737\right)^{n-1}\right] = 0.995$. This shows that the grid size is set such that a mobile in the corner will almost always have another mobile within reasonable range.

3. Packet Description and Management Processes

This section provides a more detailed description of the simulated packet and discusses general packet management processes associated with both the COTM link and the ad-hoc link. These packet management processes include packet generation, tracking and control.

a. Packet Description

The baseline simulation assumes a packet size of 1024 symbols for determining total transfer time. The simulation does not actually generate, track or control an actual full-size data packet. Instead the simulation represents a data packet (baseline of 1024 symbols) as a ten-item vector or a row within a M × 10 matrix (where the matrix represents a buffer or queue). While all packet vectors are the same length, they do not contain the same information because different information is required depending on purpose and the link involved. There are four different packet vectors as shown in Table 18.

Table 18. Packet Header Contents.

	Ad-Hoc packet	Ad-Hoc ACK	COTM packet	COTM ACK
Entry 1	Packet type	Packet type	Packet type	Packet type
Entry 2	Track number	Track number	Track number	Track number
Entry 3	Duration	Duration	Duration	Duration
Entry 4	Origin	Origin	Origin or destination	Origin or destination
Entry 5	Destination	Destination	Generation time	Generation time
Entry 6	Current loc	Current Loc	Time pending	Time pending
Entry 7	Generation time	Generation time	Rx start time	Rx start time
Entry 8	Time pending	Original packet time pending	Rx stop time	Rx stop time
Entry 9	Time acknowledged	Original data packet number	Time acknowledged	Original data packet number
Entry 10	No. of transmit attempts so far	ACK or NACK?	No. of transmit attempts so far	Time data packet pending

The different values that may comprise a packet vector are explained below:

1. Packet type: A single digit identifies a packet as either data packet or acknowledgement and for which link (uplink, downlink or mobile ad-hoc link). Model 1 has three types of data packets and three types of acknowledgements. The three data packet types are 2, 5, 8 for ad-hoc, uplink and downlink, respectively. The three acknowledgement types are 3, 6, and 9 for ad-hoc, uplink and downlink, respectively. Model 2 adds two more acknowledgement types but is otherwise the same. The extra two acknowledgement types are type 4 and 7. These are reserved for acknowledging uplink or downlink packets transferred on the mobile ad-hoc link by the Neighbor protocol.

2. Tracking Number: The tracking number represents a sequential number indicating the order of packet generation. The sequences are independent for the mobile ad-hoc link, COTM uplink and COTM downlink. When combined with packet type, the combination provides a unique address for any single packet. It is not unique with respect to multiple copies of that same packet, as generated by the time-out process (see "Lost packets" below). To identify a specific instance of a unique packet, we need to add at least one more piece of information, like "Time pending".

3. Duration: The packet duration is measured in TDMA time slots and is entirely defined by the packet type. Data packets are currently set to a duration of 8 time slots where each time slot is 128 symbol durations and total packet length is 1024 symbols. Acknowledgements are currently set to a duration of 1 time slot or 128 symbols.

4. Origin: The mobile ad-hoc routing protocol keeps track of a packet's origin primarily for trouble-shooting via manual packet tracing. The same is true for uplink packets which do not require a destination entry (see 5 below).

5. Destination: All packet types except the uplink packets have a destination value which allows the protocols to determine when a packet has been successfully delivered. The uplink packets do not need a destination entry because they all have the same destination, the satellite. Instead, uplink packets have the origin mobile in this vector location. In a real-world implementation of the protocol, the satellite would not be a final destination and uplink packets would still need a destination address. For the simulation, destination addresses are a single digit representing the number of the mobile that is the final destination.

6. Current location: This field is only maintained for the ad-hoc packets. This field keeps track of a packet's current location in the network, which may not be the origin or destination. This field is important for accurate maintenance of the LRR metrics.

7. Generation time: All data packets have a generation time. This is used to determine a packets age and develop metrics on protocol performance by comparing the difference between time generated, time transmitted, and time acknowledged. More than one packet can be generated during any single time step, so this parameter is necessarily unique for any particular packet. When the *time-out* process regenerates a lost packet, this value is not reset. Instead it is maintained as the original generation time.

8. Time Pending: All packets have a time pending. This represents the time step in which the packet has finished transmission and is moved to the pending queue to wait for acknowledgement. This time stamp is used by the time-out process to determine when a packet is probably lost and needs to be regenerated (see discussion "lost packet"). This value is updated when the time-out process regenerates a lost packet and it is retransmitted.

9. Time Acknowledged: Only data packets have an entry for capturing the acknowledgement time. This purpose of this field is collection of performance metrics. Round-trip packet latency can be determined by taking the difference of the acknowledgement time and generation time. This field is empty when the packet is transmitted and shifted from the active queue to the pending queue. The field is completed when the acknowledgement arrives and the packet is removed from the pending queue.

10. Number of transmit attempts so far: Only data packets have an entry tracking the number of transmit attempts. This information is maintained for protocol

performance analysis and comparison. A large number of attempts to successfully transmit a packet implies a relatively poor performance with the possibility of network congestion.

11. ACK or NAK: This field is only found in ad-hoc network acknowledgements. The field has a value of either zero or one. A value of one indicates successful receipt of the data packet. A value of zero indicates that the data packet was damaged beyond the capabilities of the FEC code and will need to be transmitted. This is a key step in the selective-reject ARQ process (discussed later in Section 3.*c*).

12. Rx start time: This field represents the time step in which the packet begins to arrive at the receive (Rx) end of the COTM channel. The arrival time is equal to the start of transmission plus the one-way path delay. If the packet was transmitted by the satellite and the intended mobile is blocked at Rx start time, then the packet is lost.

13. Rx stop time: This field represents the time step in which the last symbols of the packet arrive at the receive (Rx) end of the COTM channel. The arrival time is equal to the end of transmission plus the one-way path delay. If the packet was transmitted by the satellite and the intended mobile is blocked at Rx stop time, then the packet is lost.

b. Packet Generation

As discussed [27], packet generation tends to come in bursts. The Markov model that is prevalent throughout the simulation is also an appropriate method for simulating a bursty packet generation process. The implementation used for the simulation is an $n+1$ state Markov model where n is the number of mobiles. State 0 represents a time step without a new packet being generated. States 1 through n represent time steps in which a new packet is generated and the specific state identifies a specific destination. States 1 through $(n-1)$ each represent a neighboring mobile destination and State n represents the satellite. When the Markov packet generation process transitions to a state other than State 0, the model generates a packet destined for the associated destination. For model simplification, transitions are currently limited to being between State 0 and all other states. This means that a burst of packets is followed by at least one time step without a new packet being generated. While this assumption is not required, it simplifies the solution and associated computations.

The behavior of this model is completely influenced by the transition probabilities a_{ij} which together form the Transition Matrix. A graphical representation of the model is shown in Figure 42. This graphical representation assumes that the source (transmit) node is mobile 1 and that the destinations are mobiles 2 through n and the satellite.

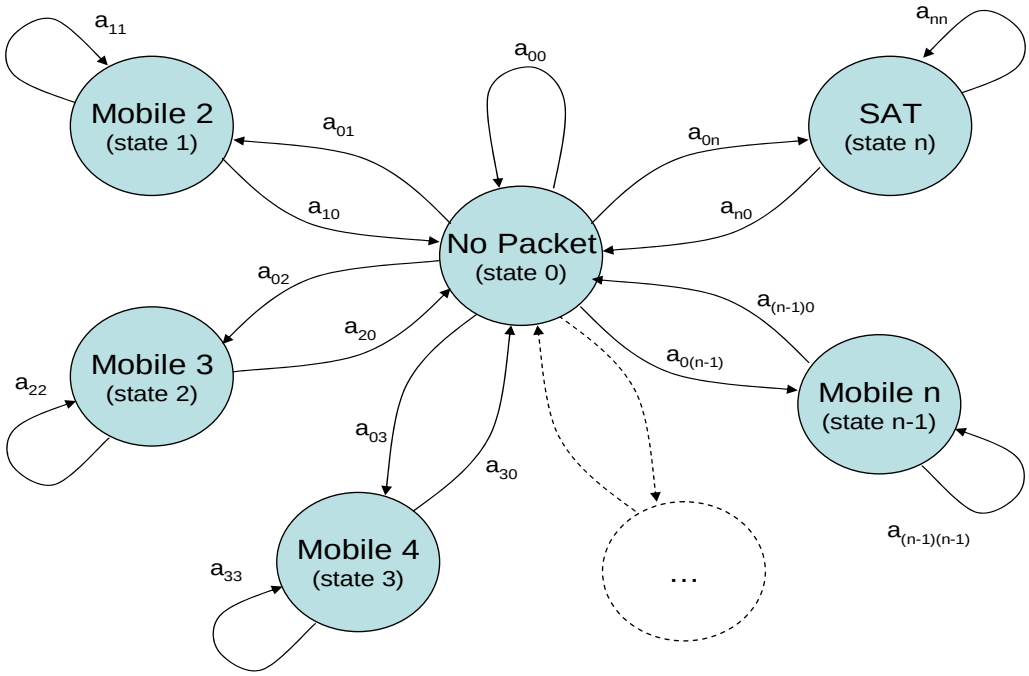

Figure 42. Markovian packet generation model.

The complete transition matrix $\mathbf{A}$ is $n+1$ by $n+1$, but several transition probabilities are zero due to the restricted assumption on transitions between non-zero states. The impact of this assumption can be seen from

$$\mathbf{A} = \begin{bmatrix} a_{00} & a_{01} & a_{02} & \cdots & a_{0n} \\ a_{10} & a_{11} & 0 & 0 & 0 \\ a_{20} & 0 & a_{22} & 0 & 0 \\ \vdots & 0 & 0 & \ddots & 0 \\ a_{n0} & 0 & 0 & 0 & a_{nn} \end{bmatrix}. \tag{3.9}$$

This leaves $(3n+1)$ transition probabilities to compute. The diagonal axis of the matrix a_{ii} represents the probability of remaining in a state once that state is achieved. This in turn impacts the interarrival times or burst statistics. As discussed in Chapter II, the interarrival times are an exponential random process of the form shown in Equation (2.1). An additional simplification benefit of assuming no transitions between States 1 through n is that States 1…n behave like independent two-state Markov models. This makes Equation (2.2) applicable for computing probabilities for packet bursts of duration k. Likewise, the mean packet burst duration is also related to the transition probabilities a_{ii} by

$$\overline{k}_i = \frac{1}{1-a_{ii}}, \; for\ i = 1...n. \tag{3.10}$$

Even though we are implementing a Markovian packet generation process, the overall packet generation rate is still an important simulation input parameter. We need to be able to compute a transition matrix $\boldsymbol{A}$ that produces the desired long term packet generation rate r. For this particular simulation, we establish two independently controllable packet rates r_{sat} and r_{adhoc} because we are studying the satellite channel and mobile ad-hoc network as separate but interacting systems. To control the packet generation model and achieve the desired long-term packet generation rate, we need to control the Markov model's stationary probabilities $\boldsymbol{\pi}$. The vector $\boldsymbol{\pi}$ represents the k-step transition probability of the Markov process as k approaches infinity in the limit:

$$\lim_{k\to\infty} a_{ij}^{(k)} = \pi_j, \; for\ i, j = 0...n. \tag{3.11}$$

As shown in [25], the stationary distribution $\boldsymbol{\pi}$ is the solution to the following system of equations:

$$\pi \mathbf{A} = \pi, \tag{3.12}$$

$$\boldsymbol{\pi}\mathbf{1} = 1 \tag{3.13}$$

where $\boldsymbol{A}$ is the Transition matrix mentioned earlier. The second equation is another way of representing the requirement that the total sum of stationary probabilities is equal to 1. The symbol $\mathbf{1}$ represents a vector of ones. With Equations (3.10), (3.12), and (3.13), we can establish desired values for $\boldsymbol{\pi}$ and $\overline{k}_i$, and then solve for the remaining transition

probabilities of $\boldsymbol{A}$. Our selection of $\boldsymbol{\pi}$ and $\bar{k}_i$ establishes mean packet generation rates and mean packet burst durations, respectively. With a basic assumption that the distribution of ad-hoc packet destinations is uniform, we have:

$$\pi = \begin{bmatrix} 1 - r_{adhoc} - r_{sat} \\ \frac{r_{adhoc}}{n-1} \\ \vdots \\ \frac{r_{adhoc}}{n-1} \\ r_{sat} \end{bmatrix}, \tag{3.14}$$

where Equation (3.9) is applied to compute π_1. Combining (3.10), (3.12) and (3.14), we can then establish individual equations for computing the transition probabilities a_{0j} for $j = 0, \ldots, n$:

$$a_{00} = \frac{1 - r_{sat} - r_{adhoc} - \frac{r_{sat}}{\bar{k}_n} - \sum_{j=1}^{n-1} \frac{r_{adhoc}}{\bar{k}_j (n-1)}}{1 - r_{sat} - r_{adhoc}}, \tag{3.15}$$

$$a_{0j} = \frac{\pi_j}{\pi_0}\left(1 - a_{jj}\right) = \frac{r_{adhoc}}{\left(1 - r_{sat} - r_{adhoc}\right)(n-1)}\left(\frac{1}{\bar{k}_j}\right) \text{ for } j = 1..(n-1), \text{and} \tag{3.16}$$

$$a_{0n} = \frac{\pi_n}{\pi_0}\left(1 - a_{nn}\right) = \frac{r_{sat}}{\left(1 - r_{sat} - r_{adhoc}\right)}\left(\frac{1}{\bar{k}_n}\right). \tag{3.17}$$

Note that we can simplify Equation (3.15) further if we assume that packet burst traffic patterns are the same to each neighboring mobile. With this assumption, $\bar{k}_j$ becomes a constant $\bar{k}$ for $j = 1 \ldots n-1$. This simplification leads to

$$a_{00} = 1 - \frac{r_{sat}}{\bar{k}_n\left(1 - r_{sat} - r_{adhoc}\right)} - \frac{r_{adhoc}}{\bar{k}\left(1 - r_{sat} - r_{adhoc}\right)}. \tag{3.18}$$

While a Markovian packet generation process is clearly more complicated than a simple model that models packet generation events as independent and identically distributed binomial events, the Markovian approach produces more accurate short term statistics for the process. If a node generates ten packets in 100 time steps, it places more stress on the network for those packets to appear all at once than one packet every 10 time steps. The network performance on queue delay and transfer time is different for the two approaches and burst packet generation tends to have worse performance.

c. Packet Tracking and Control

Once packets are generated, the network needs to store and successfully transmit them to their final destination. While the routing protocol is a significant part of reaching that goal, there are also several supporting packet management processes that are also important. These supporting processing include: 1) deletion of old packets, 2) retransmission of damaged packets, 3) regeneration of lost packets and 4) buffer overflow management.

(1) Deletion of old packets. While a traditional network may delete stale packets after some duration, this simulation does not replicate that function. Instead, packets are stored and repeatedly transmitted until they are successfully acknowledged. This approach is an important facet of this particular form of the simulation because the goal is to compare protocol performance. Deleting old packets places an artificial upper limit on packet transfer attempts and transfer times. If performance degrades to the point that packets are taking too long and being deleted due to age, then useful, comparable performance results are no longer available.

(2) Retransmission of damaged packets. Another key aspect of the referenced routing protocols [53, 54] and the presented simulations is the use of active acknowledgements with a *selective-reject* version of automatic repeat request (ARQ). This type of ARQ has the highest throughput for a packet routing protocol and the extra complexity over *go-back-N* is not significant in today's systems. The additional complexity and overhead (e.g., buffer size, larger sequence number, more overhead

symbols) makes even more sense when long propagation delays are involved [48]. By comparing the theoretical utilization, or efficiency, of go-back-N and selective reject, we can determine the relative improvement provided by selective reject. For this comparison, time is normalized to a single packet transmit duration (i.e., transmit time = 1) and a is the one-way propagation time. The utilization U of each ARQ method is dependent on the relative window size W as compared to the packet transmission time plus round-trip propagation time [48]. To maximize utilization, we require $W \geq 2a+1$. Since the maximum window size is a function of sequencing bits n, we can simplify the comparison to focus on the relationship of a to n: $2^{n-1} \geq a+1$. By using the propagation and transmission times from the simulation, we can insert a value of 122 for a and solve for $n \geq 8$. Based on these assumptions, we now have a relationship for channel utilization for each ARQ method [48]:

$$U_{(\text{go-back-n})} = \frac{1-P}{1+2aP}, \tag{3.19}$$

$$U_{(\text{selective reject})} = 1-P, \tag{3.20}$$

where n = number of symbols assigned to packet id and sequencing,
W = maximum window size = $2^n - 1$,
a = propagation time / transmit time = 0.125 sec / 0.001024 sec = 122.07, and
P = probability that a single packet is in error and must be resent.

A ratio of these two utilization equations provides a relative measure of the channel utilization:

$$\frac{U_{(\text{selective reject})}}{U_{(\text{go-back-n})}} = 1+2aP \approx 1+244P. \tag{3.21}$$

Unless the packet error rate is extremely small, Equation (3.21) shows a substantial improvement in channel utilization for selective-reject ARQ. While [48] specifically focuses on symbol errors causing a packet resend, the urban LMSC has

a large number of lost packets and acknowledgements which would also factor into ARQ activity. When the definition of P is expanded to include the probability of losing a packet due to channel fade,

$$P = P_{error} + P_{fade}, \tag{3.22}$$

we can expect P to increase and the benefits of selective-reject to grow. Expected values of $(1 - P_{fade})$ are found in Figure 35.

It is difficult to find a longer propagation delay than that between a ground user and a geosynchronous satellite. In some versions of selective-reject ARQ, the receive node only positively acknowledges every n^{th} packet where n is usually a small integer. This model assumes positive or negative acknowledgement of each and every packet. This does increase the network overhead and additional study may be able to optimize this relationship to increase overall throughput.

While expected performance of the COTM link is substantially better with selective-reject ARQ, the local ad-hoc links do not experience the long delays inherent in the SATCOM link, and the benefits of selective reject are not significant. For ease of integrating the two networks, the simulation uses selective-reject ARQ for both.

(3) Regeneration of lost packets. Closely related to the ARQ process discussed above is the *watchdog timer* (WDT) or *time-out* function. These processes cover the situation when a packet or an acknowledgement are lost for any reason. If either the original packet or its acknowledgement are lost, then the transmitter will, after a specified amount of time, need to assume that something has gone awry and the packet should be transmitted again. When this situation occurs, the routing protocol increases the LRR metric between the transmit and receive nodes by a constant term c_a. The purpose of this LRR metric modification is to show an increased resistance between those two nodes not otherwise reflected in the LRR metric since the acknowledgement was never received and the LRR metric was not updated. When the additional packet copy is generated and transmitted, the *number of attempts* term of the packet header is incremented by one for performance analysis purposes. The new copy of the packet will also have a different *time-pending* entry than its predecessors.

While optimal watchdog timer durations are not a primary focus of this research, it was necessary to investigate the impact enough to confirm that an inappropriately set watchdog timer was not eroding network performance and otherwise hampering the protocol comparison. A short parametric study was completed to search for performance trends. All aspects of the simulation were held constant with the exception of the duration of any time-out functions. We submit that further optimization is possible as an area of future work. It is important to note that the parametric study showed that optimal time-out durations were different for the two simulations. This is not surprising since the COTM packets will certainly require more transfer time if they have to move through the ad-hoc network in addition to the uplink/downlink process. Results of this short parametric study are found in Chapter IV.

Schodorf also used a time-out function in the original investigation and varied time-out durations for performance comparison [5]. While he appears to use much longer durations, 1.5 and 3.5 sec, it is important to remember that Schodorf's investigation included a packet execution time of 1.0 seconds which is not modeled in our simulations.

The ideal time-out function is set longer than the total transmission time for the large majority of packets. Total transmission time is defined as the sum of all queue wait times and round-trip transmit times. At this setting the number of unnecessarily regenerated packets is kept to a minimum and network congestion is minimized to those packets that are actually lost and not just taking too long. In the end, the ideal time-out is a function of network congestion and, therefore, a function of packet generation rates and fade statistics. An optimal watchdog timer (WDT) would need to be adaptive, but they usually are not. The simulation does use a different WDT for Models 1 and 2 but does not currently adapt them to packet generation rate or fade statistics.

The regeneration and retransmittal of unacknowledged packets is vital to implementing a robust network but can also cause significant congestion. As mentioned earlier, the duration of the time-out threshold is an important factor in determining the amount of congestion, but most reasonable values have some balance. If

a packet is truly lost to the network, then it is valuable to regenerate it as soon as reasonable. But the other side of this situation is the regeneration of packets that are simply delayed in a queue somewhere. To minimize the congestion impact of a time-out regeneration process, it is important to add some additional packet management processes to the packet processing functions. For the most part, these type of processes are focused on not generating or deleting extra copies when the network has relevant side information. One example is for a destination node that has already received a packet but has not yet sent the acknowledgement (e.g., due to a long queue delay). If it receives a second copy of the packet, it should not generate a second acknowledgement. From the other perspective, if a transmitting node regenerates a second copy because it has yet to receive an acknowledgement but then receives the acknowledgement before actually retransmitting the packet, then it should delete the packet and not generate unnecessary congestion. The simulation implements both of these types of processes to minimize the congestion impact of the time-out function.

(4) Buffer overflow management. Like any real-world, finite traffic system, a network has a performance limit. It can process packets at some speed, and if packets are being generated or regenerated faster than they can be transmitted and acknowledged, a portion of the network may become saturated and some transmit buffers may overflow. For purposes of memory consumption, this simulation does assume a finite buffer size for each node which also mirrors any real-world implementation. If the buffer becomes full with active, waiting packets, the buffer must either overwrite packets already in the queue or reject new packets. Based on the concept that the oldest packet in the queue is getting stale, the simulation prioritizes the new packet higher and overwrites the oldest active packet in the queue. While this may be the best alternative under the circumstances, it is still a failure of the network to perform. Chapter IV results show "Packet Success Ratios" for the various models and packet generation rates, which reflects the loss of active packets to being "overwritten" among other things.

B. MODEL 1A: "STOP AND WAIT" PROTOCOL

While Model 1 represents a baseline or control case for comparison, Model 1A represents the first attempt to improve network performance by modifying the packet management protocol. This version of the network simulation includes a simple and minor modification to the COTM portion of the routing protocol: inclusion of a transmission delay or wait state at the satellite transmitter whenever the satellite receiver detects loss of carrier from a mobile user. The basic concept of this protocol modification is to take advantage of the side information available to the satellite. Uplink signal strength, even without modulated data, is an excellent indication of the COTM link state approximately t_d in the past, where t_d is defined as the duration of the mobile-to-satellite path delay. As discussed in Section A.1.b, t_d ranges from 0.l194 to 0.1390 seconds and is set to 0.125 seconds in the simulation. Depending on the fade and connection dwell time statistics, there may be a high probability that the mobile node is still in the same state that it was t_d seconds ago, but we need to keep in mind that there is an equal delay between the satellite's transmission and the arrival at the mobile. So the more important question is whether the mobile will be in the same state $2t_d$ seconds after establishing the side information. The satellite has access to this side information at the mid-point in the cycle. For the empirical data discussed in Chapter II, the probability of the mobile still being in the connection state after $2t_d$ seconds is ~90.37% as determined through linear interpolation of the CDF shown in Figure 11. Similarly, the probability of the mobile still being in the fade state is ~76.40% as shown in Figure 12. The same probabilities can be computed for the simulation's Markovian visibility model, where the complementary CDF is a geometric random variable [5]:

$$\Pr\left\{n > t_d \,\middle|\, s(m)\ldots s(m+n) = connection\right\} = a_{11}^{2t_d/\Delta t} = 0.99997^{250\,ms/0.128\,ms} = 0.9431, \quad (3.23)$$

$$\Pr\left\{n > t_d \,\middle|\, s(m)\ldots s(m+n) = fade\right\} = a_{00}^{2t_d/\Delta t} = 0.999936^{250\,ms/0.128\,ms} = 0.8825, \quad (3.24)$$

where s(t) is the discrete state of the Markov process at discrete time t,
Δt is the discrete time step of the Markov process, and
a_{ii} is the probability of transitioning from state i back to state i.

The probabilities shown above are for the state transition probabilities used to generate results presented in Chapter IV. As discussed in Chapter II, the Markovian model overestimates the dwell times for short durations when compared to the empirical data.

Model 1A's packet management protocol is based on an additional assumption beyond those listed in the introduction to Chapter III. The extra assumption is that the mobiles will maintain some kind of uplink carrier signal during their TDMA time slot, even when they do not have any data to transmit. This assumption is critical to the implementation of this traffic management protocol because the lack of signal strength needs to imply a mobile fade condition not just a lack of uplink packets.

The Model 1A simulation implements a small change to the protocol by putting the satellite transmission into a temporary wait state whenever it fails to detect an uplink carrier from a mobile. The wait state only applies to the mobile that appears to have no signal. The satellite continues to communicate with any mobile that has a detectable uplink signal. If the satellite enters a wait state associated with a specific mobile, it stays in that wait state until a fixed timer t_w counts down or until the uplink carrier reappears, whichever comes first. As the wait state duration increases, the likelihood of the latter option grows. If the wait state duration gets large enough, this protocol evolves into a cycle where the satellite downlink matches the time delayed uplink (or "lags the mobile uplink by the path delay").

As mentioned above, the impact of the Model 1A protocol modification is dependent on the relative duration of t_f as compared to t_d and t_w. This relationship can be separated into five macroscopic situations and each situation is discussed below. Figures 43 through 46 provide a helpful graphical perspective. Each of these figures has two separate horizontal axes representing time. The top time axis represents the activity at the mobile transceiver and the bottom time axis represents activity at the satellite. The separation between these two time axes represents the path delay t_d. Arrows that connect the two time axes represent the transfer of information (e.g., packet, loss of carrier) from one end of the channel to the other (which takes t_d seconds). The fade duration t_f is

shown on the mobile's end of the channel. Similarly, the wait state duration t_w is shown on the satellite's end of the channel. The transmission of packets to a blocked mobile is a network inefficiency and is shaded yellow. The lack of packet transfer to a clear mobile is another network inefficiency and is shaded blue. The goal of the Model 1A protocol is to avoid packet transfer to a blocked mobile. These regions are shaded green.

Figure 43 shows the situation where $t_f = 2t_d + t_w$. This situation is about the best that can be accomplished without fade prediction. All packets transmitted during $t_0 \pm t_d$ are blocked and require retransmission. The inefficiency associated with this period is a direct consequence of the path delay and is only avoidable through fade prediction. At $t_0 + t_d$, the satellite detects a loss of uplink carrier and enters a wait state. In this optimal case, the wait state ends t_d seconds before the fade and transmitted packets arrive just as the mobile becomes clear again. This precise situation will be very rare due to random variation in the fade duration and most fades will fall into one of the other categories. The remainder of this discussion focuses first on those situations where $t_f < 2t_d + t_w$ and then on situations where $t_f > 2t_d + t_w$.

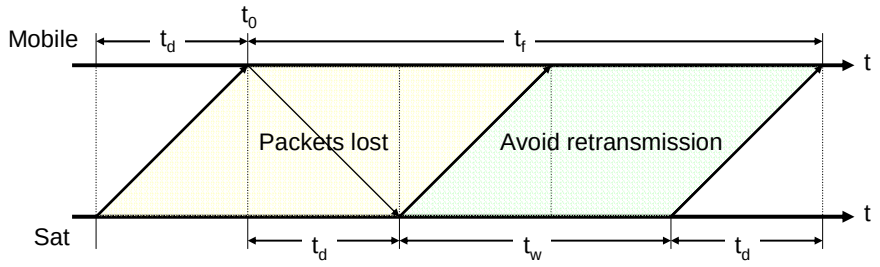

Figure 43. Packet transfer perspective for Model 1A: $t_f = 2t_d + t_w$.

Figure 44 represents the situation where the fade duration is shorter than the wait state plus the round-trip path delay but longer than either by itself. This situation introduces additional inefficiencies associated with the mobile being in a clear state but the satellite being in a wait mode. This additional inefficiency is represented by the blue-shaded section of Figure 44 (also labeled "Lost Opportunity").

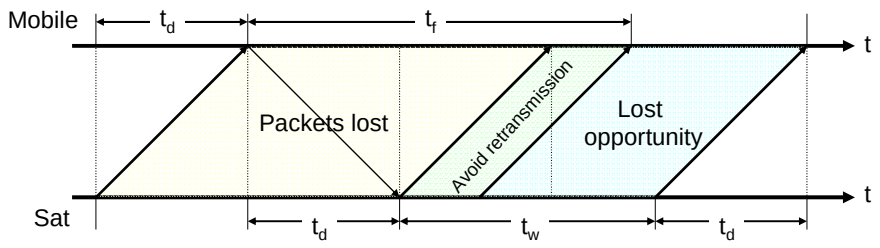

Figure 44. Packet transfer perspective for Model 1A: $t_w < 2t_d < t_f < 2t_d + t_w$.

If the fade duration is even shorter it may become shorter than the maximum wait state duration or the round-trip path delay. Each of these situations has a different impact on packet activity. If $t_f < t_w$ then the satellite receives an uplink carrier while it is still in the wait state. The appearance of an uplink carrier provides the satellite with side information that suggests that the mobile is clear and ready to receive packets. This shortens the inefficiency represented by the blue-shaded region by $t_w - t_f$ seconds. The second situation occurs when $t_f < 2t_d$. In this situation, some of the downlink packets transmitted before the satellite enters the wait state are successfully received by the mobile as it leaves the fade state. This shortens the inefficiency represented by the yellow-shaded region by $2t_d - t_f$ seconds. Figure 45 shows the combined situation where both of these criteria are true.

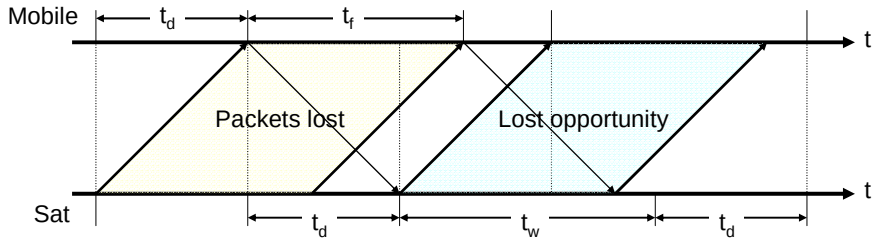

Figure 45. Packet transfer perspective for Model 1A: $t_f < 2t_d$ and $t_f < t_w$.

The last situation occurs when the fade duration is longer than the combined wait state and round-trip path delay. Figure 46 provides a graphical view of this situation. In

this situation, the mobile is blocked for the duration of the wait state. The Model 1A protocol provides increased channel efficiency by not transmitting packets for the duration of the wait state, but the wait state ends early relative to the fade. In this case, the satellite restarts transmission to a blocked mobile. These packets are lost and require retransmission at a future time.

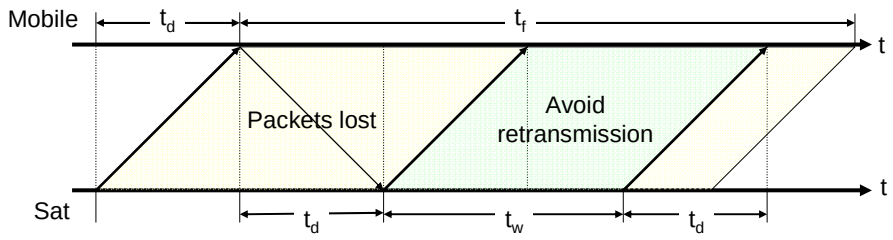

Figure 46. Packet transfer perspective for Model 1A: $2t_d + t_w < t_f$.

Figures 43 through 46 highlight the complexity of the Model 1A protocol's impacts on COTM channel performance. Depending on the specific relationship of the fade duration and the wait-state duration, the protocol can improve (green-shaded regions) or hamper (blue-shaded regions) channel efficiency. Chapter IV presents simulation-based performance estimates for several different wait-state durations.

C. MODEL 2: NEIGHBOR ROUTING PROTOCOL

Model 2 represents an investigation into mitigating the urban LMSC fade impacts by exploiting an expanded network and circumventing the problem. Like Model 1, this model is a simulation of the COTM channel and a mobile ad-hoc network, but for this model the two networks are integrated into a single seamless network.

For the most part, this model implements the packet routing protocols already presented earlier in this Chapter. Important differences are discussed below.

1. COTM Uplink Packet Routing

In the baseline COTM packet routing protocol presented in Section A, the uplink utilizes nearly real-time side information regarding channel state. When the COTM uplink enters the fade state, packets destined for the GIG are held in the buffer until the system returns to a connection state. As presented in Chapter IV, the performance of this simple protocol leaves room for improvement.

In Model 2, the network routing protocol has access to another path through the mobile ad-hoc link. When a mobile's COTM channel is in the fade state, packets destined for the COTM channel may be re-routed to a neighboring mobile for surrogate uplink. The blocked mobile requires side information to determine whether a viable neighbor is available or, if there are more than one, which one to utilize. The first decision factor is the proximity and link quality of possible neighbors. This information is already available from maintenance of the ad-hoc network and is discussed as part of the ad-hoc forwarding protocol. The second decision factor is the expected packet service time of those same neighbors, which for this simulation is focused on queue wait times. This information is also maintained as part of the ad-hoc network forwarding protocol. The last decision factor is COTM channel state of the neighboring mobiles. This information is an important routing criteria because it is of no help to send packets from one blocked mobile to another. COTM channel state is not a piece of information that is currently maintained as part of the ad-hoc network activity. The implementation of the Neighbor protocol requires an additional high frequency, cyclic data transfer between neighboring mobiles so that accurate decisions can be made.

Another important facet of the uplink Neighbor protocol is the control and use of assigned TDMA slots. One possible approach to implementing the Neighbor protocol is to transfer uplink packets to a neighbor and expect that neighbor to add those packets to its existing uplink queue. This approach has several drawbacks. The first drawback is the fact that the packet has to wait its turn behind the neighbor's existing uplink packets and transfer is delayed. This in turn increases the probability that the neighbor becomes blocked and is not be able to act as a surrogate. The second drawback is that the

surrogate's own packets are delayed while processing a neighbor's workload. And third, the COTM uplink channel is not used efficiently with this approach. Instead, a better solution is for the neighboring mobile to utilize the blocked mobile's TDMA slot while it is not capable of using it itself. This overcomes the drawbacks presented above. The neighbor can send the blocked mobile's packets during the blocked mobile's TDMA time slot and not interrupt or slow its own uplink activity. The channel is utilized more efficiently and increases the chances of getting both mobiles' packets transmitted before the channel enters the next fade state. The obvious downside of this approach is the additional complexity; and precise control of TDMA time slot ownership is crucial to its success. If a blocked mobile passes from behind a building and the channel transitions to the connection state, that mobile could begin transmitting, but if it has passed temporary control and use of its TDMA slot to a neighbor, they may transmit on top of each other and effectively destroy both transmissions. In the simulation, control of the TDMA slot is treated like a token that only one mobile can hold at a time. When a blocked mobile transfers its uplink packets and control of its TDMA time slot to a neighbor, it cannot restart uplink transmission until the neighbor has returned control of the TDMA time slot. The average fade duration is an important factor in determining whether this approach is an overall improvement. In the situation where a fade is relatively short, a blocked mobile may start to transfer packets and TDMA slot control to a neighbor and then become clear. In this case, the mobile is not able to use its own uplink until that neighbor has finished packet uplink and returned control of the TDMA resource. If the average fade duration is shorter than the average time to transfer uplink packets through a neighbor, then the Neighbor protocol hampers channel performance instead of helping. The results presented in Chapter IV for the "Boston" and "New York" environments show that this is not the case for a generous range of packet generation rates. Yet, as the ad-hoc network approaches congestion, the uplink packet transfer times grow and the Neighbor protocol performance degrades to the point of being worse than the Model 1 baseline.

The secondary routing path through the ad-hoc network adds several steps to the packet delivery process, so it is important that these steps not be overly time consuming.

This leads to the development of three constraints on the use of this process: 1) the blocked mobile's ad-hoc queue needs to be nearly empty to keep packet service time to a minimum, and 2) the link quality with the neighbor needs to be high, and 3) the ad-hoc queue size of a neighbor needs to be low.

The actual sequential implementation starts with the blocked mobile assessing whether its own mobile ad-hoc buffer is fairly empty of packets, which suggests that the uplink packet would be transferred to a neighbor fairly quickly. If this is not the case, then the mobile just holds the uplink packets in the uplink buffer and waits for the COTM channel to change back to the connection state or the mobile ad-hoc buffer to become less congested. If the mobile ad-hoc buffer has less than $N_{threshold}$ packets, then it continues to investigate the viability of utilizing a neighbor by checking to see if any neighbors have COTM LOS to the satellite. If this is also true, then those neighbors who have COTM connections are compared using a LRR style metric. The metric equation is

$$LRR(i,j) = L(i,j) + \alpha U(j) + c_p u(L(i,j) > r_e), \tag{3.25}$$

where

$L(i,j)$ is the link quality metric from mobile i to mobile j,
$U(j)$ is the ad-hoc buffer utilization of mobile j,
α is a scalar weighting coefficient,
r_e is the correction threshold of the forward error correction scheme,
$u(x > t)$ is the unit step function and equals one when x is greater than t, and
c_p is a scalar constant, and serves as the penalty increase to a metric.

The link quality metric is the same link quality metric used by the ad-hoc forwarding protocol and is primarily based on recent bit error rate. The buffer utilization metric is also used by the ad-hoc forwarding protocol and is a simple ratio of active packet count vs. buffer size. It is important to note that the receiving mobile's buffer only plays a secondary role in the Neighbor protocol. The uplink packet does not spend time in the neighbor's ad-hoc queue but instead goes from the blocked mobile's ad-hoc queue to the neighbor's uplink queue for transmission on the COTM link. The satellite's acknowledgment often comes back through the neighbor's ad-hoc queue on its way back

to the blocked mobile. While this is not always the case, it does occur frequently. When the neighbor's ad-hoc queue is very congested, the acknowledgement gets delayed and the blocked mobile may regenerate extra copies when the time-out function erodes to zero. Because the utilization metric U is usually much larger than the link quality metric L a weighting coefficient is used to balance the impact. In the active simulation, this weighting coefficient is currently set to 0.333. The third and final term in the LRR metric computation is the penalty function. This extra term has the same purpose as in the ad-hoc forwarding protocol, to penalize links that are above the capability of the FEC code. This effectively increases the relative weighting of the link quality term when the link quality term is above the error rate threshold r_e. The value of the penalty that is assessed in those cases is c_p. In the simulation, this penalty is set at 0.025.

The last step of the Neighbor protocol is to determine if the neighbor with the best metric has a good enough metric to continue. The purpose of this last check is to confirm that the process has a high chance of being quick and, therefore, a minimal impact on the network. If the best neighbor has a marginal link quality or congested buffer, then it becomes more likely that the uplink packets or their acknowledgements will be delayed even longer than just waiting for a direct connection or that they will cause undue burden on the ad-hoc network with transmission failures and/or extra copies.

2. COTM Downlink Packet Routing

In the baseline (Model 1) COTM packet routing protocol, the satellite downlink does not utilize any side information regarding channel state and simply sends packets as quickly as possible. If the destination mobile's COTM channel is in the fade state when the packet reaches the mobile node (t_d seconds later for path delay), then the packet is lost. Model 1A simulates one modification to this approach. In that model, the downlink does utilize available side information to modulate packet transmission. Unfortunately, the available side information is significantly delayed due to path length. Another approach may work better.

In Model 2, the satellite downlink does not utilize any side information and reverts to simply transmitting packets as quickly as possible. Instead, Model 2 relies on the ad-hoc network for improved performance. In Model 1, the mobile receivers are only collecting data during their assigned TDMA slot. The receiver resources are not utilized during other slots.

In the Neighbor protocol, we expand the use of the receivers that have LOS to the satellite. As discussed in the previous section, the COTM uplink portion of the Neighbor protocol relies on accurate knowledge of the fade/connection state of neighboring mobiles. The downlink portion of this protocol also heavily relies on this information. In the Neighbor protocol, if a mobile becomes blocked (fade state), it relies on a neighbor to collect packets being transmitted on the satellite downlink. This does not impact that neighbor's collection of its own downlink since they are being transmitted during different TDMA slots. Like the other routing decisions involved in the neighbor and ad-hoc protocols, this decision is based on the same metric equation presented in the previous section (see Equation (3.25)). The metric equation parameters were selected through a limited trial-and-error effort; focused optimization can be accomplished in future studies. The simulation currently assumes perfect side information of COTM channel state and buffer utilization factors U for computing this protocol. In Chapter V, we recommend a future effort to remove this ideal assumption and model the impact of a realistic delay or inaccuracy in this knowledge.

3. Model 2 Summary

The Neighbor protocol provides improvements in network performance, but not without a tradeoff. It is important to understand the downside of this mitigation approach. The first is the obvious reliance on a local-area ad-hoc network and physically nearby neighbors. If a mobile node is on its own, the mitigation approach will not be of any assistance, and if there is a local area ad-hoc network to provide this circumvention, the performance of that network will degrade due to the additional burden of the Neighbor protocol. If the ad-hoc network is already running close to capacity, the

Neighbor protocol may increase congestion to the point of network inadequacy. Simulation results are presented in Chapter IV.

D. CHAPTER III SUMMARY

In this chapter, we introduced three network simulation models that are focused on achieving the second research goal presented in Chapter I, that of mitigating the impacts of the urban fade on network performance. While the primary focus of the simulation effort is the COTM channel, the simulations also include a local area mobile ad-hoc network as part of a possible mitigation strategy. The first of the three models is the baseline design which provides a standard for comparison. The second and third models represent possible mitigation strategies. In the next chapter, we analyze performance results for each of these models and assess their success at mitigating urban fade impacts.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. RESULTS: URBAN COTM AND AD-HOC MODELS

Simulation-based performance estimates for network Models 1, 1A, and 2 are presented in this chapter. Model 1 represents the basic network configuration and provides the comparison baseline. Model 1A incorporates a small modification to the COTM channel protocol and represents an attempt to reduce the number of downlink packets transmitted to a blocked mobile. Model 2 integrates the ad-hoc and COTM networks using the Neighbor protocol and represents an attempt to improve performance by circumventing the blockage. The results presented include direct comparisons of identical simulation runs (e.g., identical LOS visibility, packet generation and vehicle motion vectors) and comparison of general performance statistics. The summary performance statistics are consistently based on twenty independent process samples, where each sample is 51.2 seconds in duration.

The first three sections of this chapter present and discuss key simulation parameter settings. Section A introduces the topic of simulation parameters and provides rationale for most of the key parameters. Section B follows with an in-depth investigation of the appropriate pending packet time-out thresholds. Section C discusses estimation of process statistics and the application of the Central Limit Theorem.

The last two sections present performance results for two urban environments having significantly different densities. The low density city, which we refer to as Boston, implements a satellite connection/fade ratio of 66.6/33.3%. The higher density city, which we refer to as New York, reverses this ratio and implements a connection/fade ratio of 33.3/66.6%.

A. MODEL PARAMETER SETTINGS, ASSUMPTIONS AND RATIONALE

While Chapter III provides a general overview of how the simulation works, Chapter IV presents performance results for a single, specific, simulated network in two different urban environments. Since the goal of this research is to investigate mitigation strategies for the TCA's COTM link, the simulation parameters have been selected to

represent a small group of HMMWVs using the Joint Tactical Radio System (JTRS) to maintain a MANET and using the COTM link to communicate with the GIG through a TSAT. A list of key simulation parameters and the assigned values are shown in Table 19. The rationale for these settings are discussed below.

Table 19. Simulation parameter settings.

Parameter	Setting or Value
Number of mobiles	5 vehicles
Size of movement area	350 x 350 meters
COTM symbol duration / data rate	1 μs / 1 Msps
Ad-hoc symbol duration / data rate	1 μs / 1 Msps
Packet size	1024 symbols
TDMA slot duration (simulation time step)	128 symbols
Average connection duration (Boston/New York)	4.2 seconds / 2.0 seconds
Average fade duration (Boston/New York)	2.0 seconds / 4.2 seconds
Total mobile node packet generation rate r_p	Varies (7.8125 – 78.125 pps)
Local area (mobile ad-hoc) packet generation rate	$r_{adhoc} = 0.4r_p$
COTM packet generation rate	$r_{sat} = 0.6r_p$
Maximum queue (buffer) size	3 Msymbols
Pending packet time-out thresholds	see Section B

1. Number of mobiles: The primary rationale for limiting the simulation to five mobiles is one of computer memory and computation time. With each additional mobile, the memory requirements increase proportionally (e.g., packet queues, visibility vector, packet generation vector, and motion vectors). Simulation runs with six or more mobiles would sometimes stop execution due to an "out-of-memory" error. This limitation can be circumvented with additional computer memory. Simulation memory management issues are discussed in Appendix B.

2. Size of movement area: As discussed in Chapter III, Sections A.2.d and A.2.e, the mobile movement area is a Cartesian grid. The size of the grid is a balance between crowding the mobiles together and spreading them too thin. On the one hand, the size of the grid is chosen to minimize the probability of a mobile being out-of-range from its neighbors. The selected grid size limits this scenario to a probability of only 0.5% (see Section A.2.e). On the other hand, the size of the grid needs to be large enough to require multi-hop packet transfers so that the routing protocol is exercised. With a 350 x 350 meter grid, the maximum vehicle separation distance is 495 meters, which is approximately 50% longer than the normally successful one-hop mobile ad-hoc communication distance.

3. COTM data rate and symbol duration: The COTM link instantaneous data rate is set at one megasymbol per second (Msps). This is a reasonable estimate based on the TCA plans to offer up to 1.544 Mbps to the individual tactical user [1].

4. Ad-hoc data rate and symbol duration: The ad-hoc instantaneous data is also set at one Msps. This data rate is not currently achievable with existing mobile radios (e.g., SINCGARS) but may become available with the development of the Joint Tactical Radio System (JTRS) and its Wideband Networking Waveform (WNW). If the actual data rates available to JTRS are lower than one Msps, the congestion point of the ad-hoc network will be lower than presented in these results, but general findings and trends will still be valid.

5. Packet size: The packet size is set at one kilosymbol to minimize the probability of losing partial packets to urban blockage. In a real, deployed system, optimal packet size will be a balance between increased inefficiency due to overhead symbols and increased inefficiency due to partially blocked packets. An additional benefit of this parameter setting is the easy conversion of simulation throughput results from packets to kilosymbols.

6. TDMA slot duration: As discussed earlier in Chapter III, Section A.1, the TDMA slot duration is arbitrarily set to $1/8^{th}$ of a packet size. A longer TDMA slot duration is expected to improve network performance due to a reduction in partially blocked packets. This additional investigation is recommended as future work. It is also important to note that the simulation time step is one TDMA time slot duration.

7. Average connection duration: As discussed in Chapter III, the simulation's visibility model is a 2-state, discrete Markovian process. This makes the average connection duration a function of the connection-to-connection state transition probability a_{11}. To approximate the urban density of Boston, this parameter is set at 0.99997. To approximate a more densely populated area, like New York, this parameter is set at 0.999936.

8. Average fade duration: Like average connection duration, average fade duration is entirely determined by the fade-to-fade state transition probability a_{00}. This parameter is set to 0.999936 and 0.99997 for the Boston and New York results, respectively.

9. Total mobile node packet generation rate: The Markovian packet generation model is introduced in Chapter III. This model allows the selection of independent packet generation rates for locally destined packets and COTM channel packets. These settings are discussed as Items 10 and 11 below. When combined, they represent the total rate of new packets appearing at a single mobile node. For the total number of

new packets appearing in the simulation, we need to multiply by the number of mobiles and then add the packets appearing at the satellite node (see paragraph 10 below): $r_{total} = nr_p + nr_{sat}$.

The packet generation rates were varied as part of analyzing network performance. The range of packet generation rates is selected to be near, but generally below, the network congestion point. For the lower density urban environment, the congestion point occurs at or slightly above 78.125 pps. The high density urban environment, New York, has a substantially lower congestion point and analysis is limited to 46.875 pps and below.

10. COTM packet generation rate: The Markovian packet generation model allows independent packet generation rates for locally destined packets and COTM channel packets. The simulation results are based on a 3:2 bias towards COTM packets based on the rationale that more of a mobile's communication (in symbols per second) will be with the GIG than with immediate neighbors. This bias is supported by the network because the COTM channel has a higher capacity than the ad-hoc network. Each mobile node generates COTM packets at an average rate of r_{sat}. Similarly, the satellite node generates COTM packets at this same rate for each mobile customer. This makes the total number of new packets appearing at the satellite node nr_{sat}. These settings reflect an equal level of uplink and downlink activity. Using the packet description vector presented in Chapter III, we can identify uplink COTM packets as packet type = 5, while downlink COTM packets are identified as packet type = 8.

11. Local area (mobile ad-hoc) packet generation rate: As mentioned above, each mobile generates packets destined for local area neighbors. These packets are only generated by the mobiles and only travel on the mobile ad-hoc network. The results presented in this chapter assume that two of every five packets generated at a mobile are destined for a local neighbor. The satellite node does not generate local area packets. In Chapter III, local area packets are identified as packet type = 2.

12. Maximum queue (buffer) size: Due to real-world limitations of finite system memory, simulation packet queues are limited to three megasymbols (Msym). For the selected packet size of one ksym, this equates to a maximum of 3000 active packets in any queue. When combined with packet generation rates, this parameter helps determine whether active packets are overwritten due to congestion and buffer overflow.

B. SELECTION OF THE PACKET TIME-OUT THRESHOLD

As mentioned in Chapter III, the COTM and ad-hoc protocols include an acknowledgement time-out or watchdog timer. The purpose of this packet management

subprocess is to regenerate and retransmit packets that are lost to the network, where *lost* means that the packet transmission was not successfully received by the destination or the acknowledgement was not successfully received by the source/origin. The central decision related to implementing this subprocess is to determine how long a packet can be pending before it has to be regenerated and retransmitted. While a detailed investigation of the optimal value is not the focus of this research, it is important to confirm that an appropriate value is selected so that network performance is not inappropriately hampered by this subprocess diverting attention from the central investigation into long-duration urban signal fades.

The optimal value for timer duration is a complex balance between the desire to resend truly lost packets as soon as possible and not unnecessarily regenerating packets that are simply caught in network congestion (waiting in a queue somewhere in the network). This balance is dependent on the amount of network congestion, and network congestion is dependent on the packet generation rate. This set of relationships suggests an important optimal threshold dependence on packet generation rate. While for brevity the rest of this section presents results for a fixed packet generation rate, additional parametric studies were completed that confirmed a monotonic relationship between optimal threshold settings and packet generation rate. Lower packet generation rates had lower optimal time-out thresholds, and higher packet generation rates had higher optimal time-out thresholds. To minimize simulation complexity and keep the focus on the primary research problem, an adaptive time-out threshold was not pursued. Instead, the time-out thresholds were optimized for a moderately high packet generation rate and held constant for other generation rates. In addition to reducing simulation complexity, this approach was also crucial to providing an "apples-to-apples" comparison of model results.

There is a large difference in the time-out thresholds of the two different communication channels. The optimal ad-hoc time-out threshold tends to be substantially smaller than its equivalent for the COTM channel. The first significant difference is obviously the path delay. Assuming no packet service execution time, the

path delay is a bare minimum value for the time-out threshold since it represents the ideal minimum packet latency. A second reason for this large difference is the deep signal fade experienced by the COTM channel. For Boston, the average fade is 1.68 seconds, and we expect similar or larger values in other urban areas. This average fade duration is approximately three to five times larger than the watchdog timer settings. As the watchdog timer setting gets shorter, the downlink sends more packets that are not received by the blocked mobile. This increases network congestion without a gain in successful delivery. A third reason that affects only Model 2 is the additional COTM packet latency caused by extra stops in the ad-hoc network before uplink and/or downlink.

1. Time-out Thresholds for Model 1

To support the selection of appropriate timer values, the simulation was run multiple times holding all values constant with the exception of watchdog timer durations. The results are shown in Figures 47 and 48. The overall mobile node packet generation rate was set at 0.008 packets per time step (ppts), with a 60/40 bias towards SATCOM packets. This bias results in a SATCOM packet generation rate of 0.0048 and an ad-hoc packet generation rate of 0.0032 ppts. In Model 1, the COTM channel and the MANET are completely separate networks. This isolates the impact of their time-out thresholds to the associated channel. Figure 47 shows the packet latency and network throughput performance for the COTM channel as the COTM packet time-out threshold is varied. Similarly, Figure 48 shows the same performance parameters for the ad-hoc network as the local packet time-out threshold is varied.

It is important to analyze the uplink and downlink performance separately because their behaviors are so different due to the proximity differences with respect to the physical blockage. The mobile uplink knows almost immediately that it has lost connection to the satellite and holds packets in the uplink queue until the signal returns. This is not true for the downlink. Due to the long path delay, the satellite does not have accurate, timely information regarding link quality and will transmit regardless.

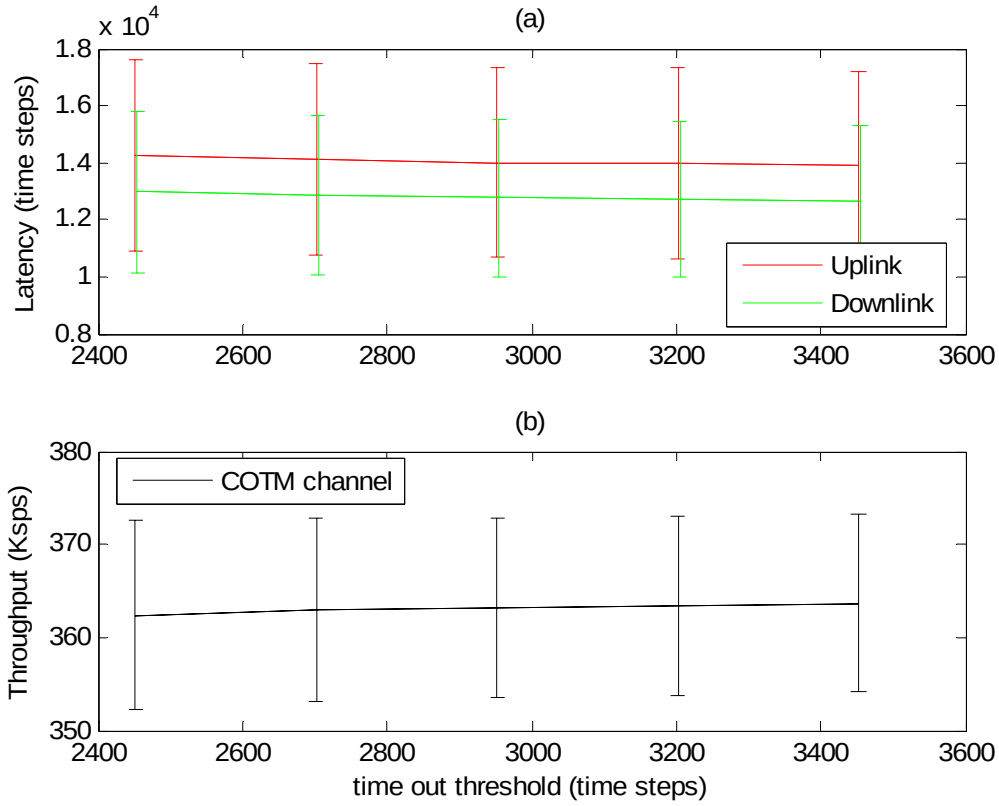

Figure 47. Comparison of COTM channel (a) packet latency and (b) total throughput as a function of COTM packet time-out threshold (mean and 95% confidence interval, packet gen. rate = 0.008).

As shown in Figure 47, COTM performance is somewhat insensitive to the time-out threshold setting. Even as the threshold increases by 40% (from 2452 to 3452 time steps), the latency decreases by only ~2.5%. Similarly, the throughput only improves by 0.4%. This shows that the channel is not near capacity and that the extra congestion caused by a short watchdog timer is easily absorbed by the channel. We would expect to find a stronger performance dependence for higher packet generation rates that would be more prone to congestion.

The ad-hoc network performance is not as insensitive to watchdog timer thresholds. Both packet latency and network throughput increase markedly until $t_{time\text{-}out}$ reaches 200 time steps. At that point performance starts to degrade slowly. This transition is presented in Figure 48.

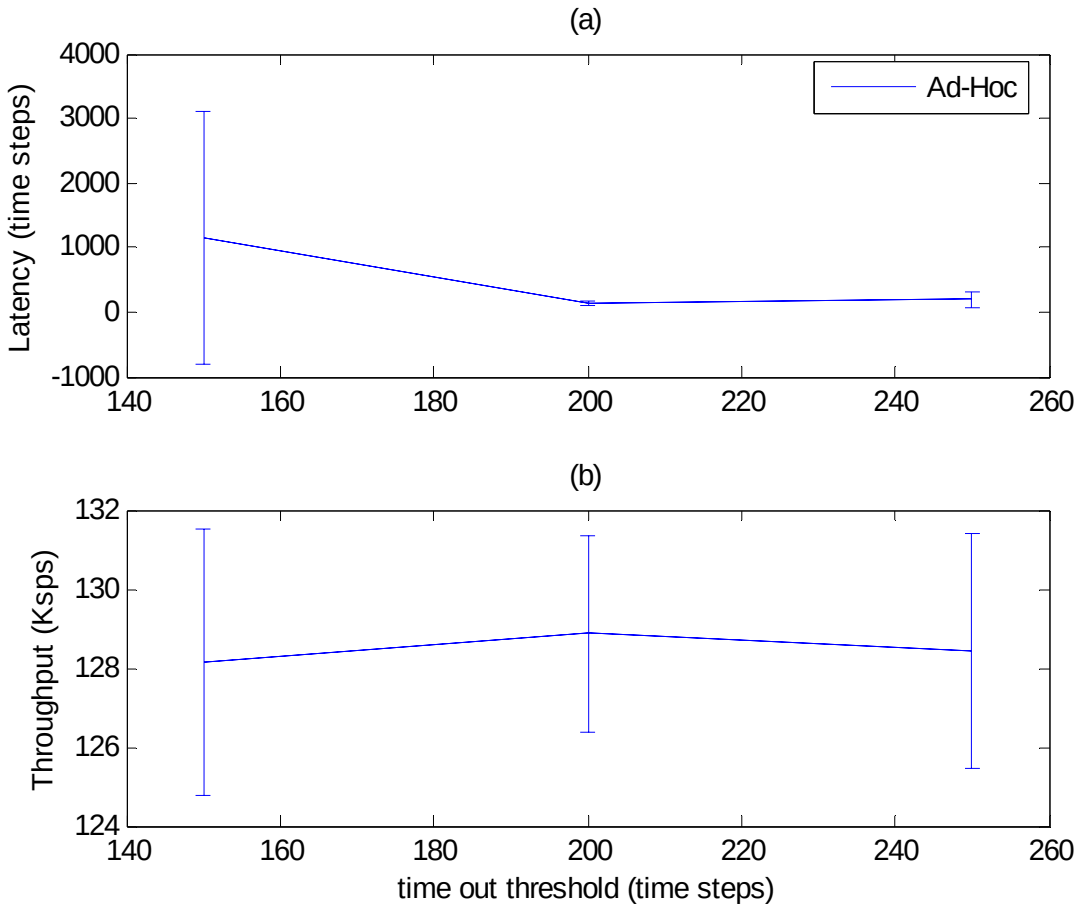

Figure 48. Comparison of MANET (a) packet latency and (b) network throughput as a function of ad-hoc packet time-out threshold (mean and 95% confidence interval, packet gen. rate = 0.008).

Based on this parametric study, we selected watchdog timer thresholds of 1250 and 200 time steps for the COTM packets and MANET packets, respectively. It does appear that another small improvement in performance is available by increasing the COTM packet threshold to 1500 time steps, but the increase would hurt performance at lower packet generation rates.

2. Time-out Thresholds for Model 2

In Model 2, the two networks become integrated, and COTM packets frequently travel through the ad-hoc network. This network integration broadens the impact of each

time-out threshold to have a ripple effect on the other channel's performance. In an effort to isolate the impacts of each time-out threshold, one was held fixed while the other was varied. For this parametric study, the packet generation rate was held constant at 62.5 packets per second per mobile (0.008 packets per time step).

Figures 49 and 50 show the network performance when the ad-hoc time-out threshold is fixed at 400 time steps and the COTM threshold is varied. Model 2 was exercised with five different settings of COTM time-out threshold: 2952, 3452, 3952, 4452, and 4952 time steps. These values translate to 0.378, 0.442, 0.506, 0.570, and 0.634 seconds, respectively. As mentioned in Chapter III, these numbers are considerably smaller than the time-out delays investigated by Schodorf, but when we subtract the 1.0 second execution time from Schodorf's 1.5 second time-out delay, we find that Schodorf's setting nearly matches our mid-point of 0.506 seconds. The five settings for this parameter appear to have non-standard values. This is due to the simulation using the roundtrip path delay, 1952 time steps or 0.125 seconds, as a base value. The roundtrip path delay is used as the base value because it represents the theoretical minimum for a time-out threshold. In an ideal network with no queue delay or execution time, this would be the correct value for the time-out threshold. Figure 49 shows network performance with respect to packet latency and Figure 50 shows network performance with respect to channel throughput. Both figures show steady improvement of network performance as the time-out threshold is increased from 2952 time steps to 3952 time steps, but as the time-out threshold increases beyond 3952 time steps the results do not clearly show any additional improvement. Instead, the latency and throughput results show similar or slightly worse performance. These same trends also appeared in the results for packet success ratio and transmit-only latency, which are not shown for brevity. The figures show the mean performance over 20 iterations of 400,000 time steps each. They also show the 95% confidence error bars bracketing that mean. The error bars are dependent on both sample variance and number of iterations completed.

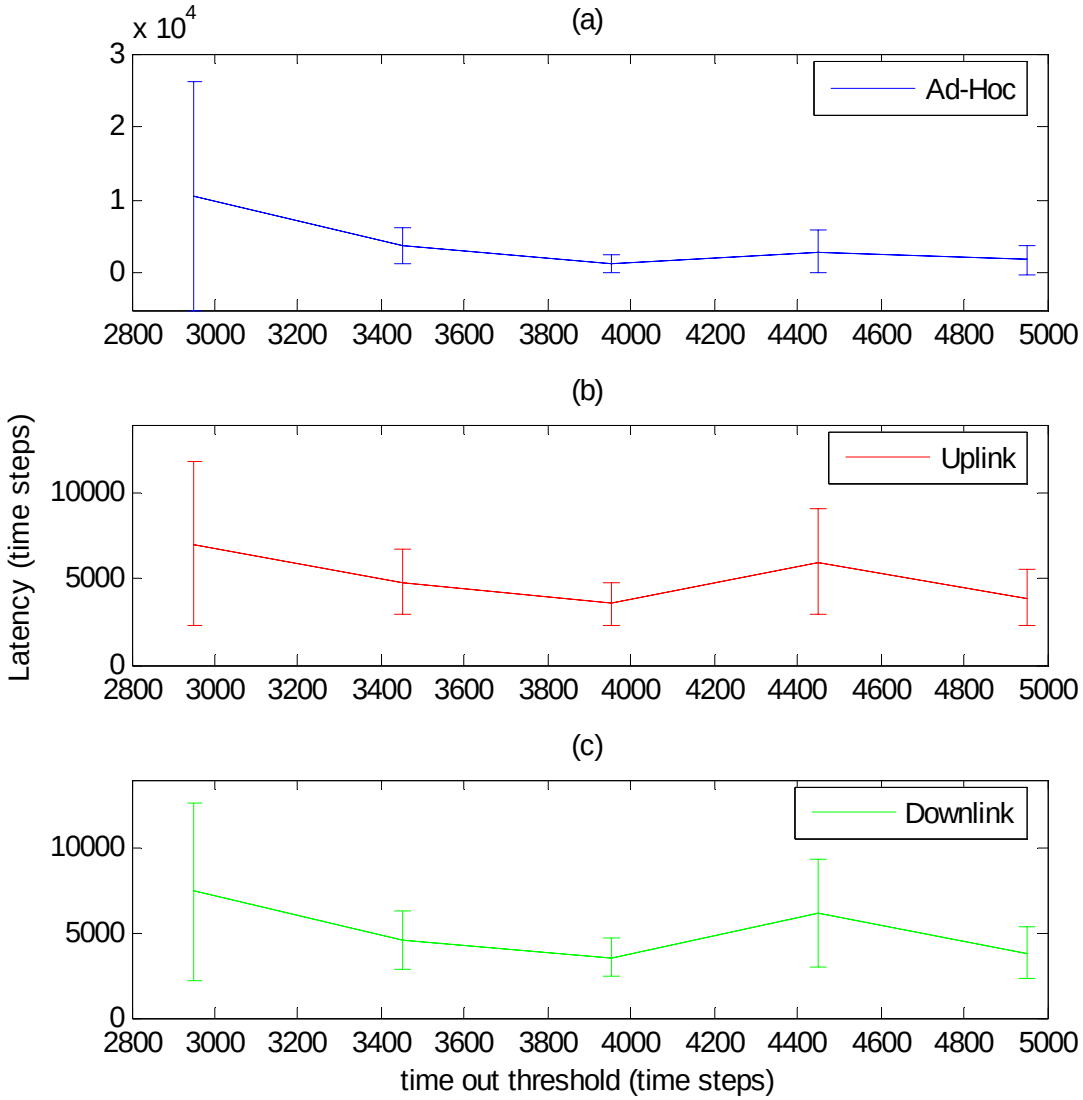

Figure 49. Comparison of packet latency (for transfer and acknowledgement) as a function of COTM packet time-out threshold for (a) ad-hoc packets, (b) uplink packets, and (c) downlink packets (mean and 95% confidence interval).

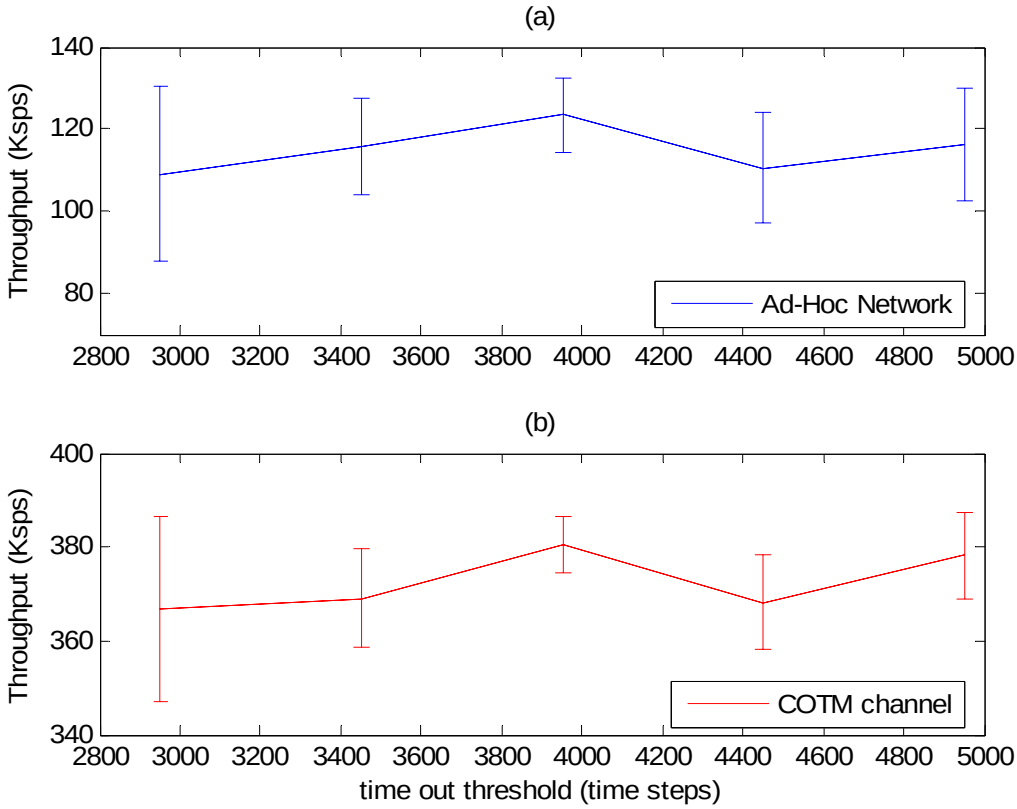

Figure 50. Comparison of channel throughput as a function of COTM packet time-out threshold for (a) ad-hoc network and (b) COTM channel, uplink and downlink (mean and 95% confidence interval).

It is important to note that the COTM time-out thresholds for Model 2 are noticeably larger than those set for Model 1. This is a reflection of the extra processing associated with this integrated network. In addition to the uplink and downlink queue wait times that are part of the basic COTM link in Model 1, COTM packets may now have additional queue wait times in the ad-hoc network for either the uplink direction, downlink direction, or both.

A similar investigation was performed with respect to the impacts of the ad-hoc network time-out settings. For this investigation, the COTM packet time-out was fixed at 3952 time steps (0.502 seconds), and the ad-hoc packet time-out was varied between 300,

350, 400, 450, and 500 time steps. These settings correspond to 0.0384, 0.0448, 0.0512, 0.0576, and 0.064 seconds, respectively. The performance results are shown in Figures 51 and 52.

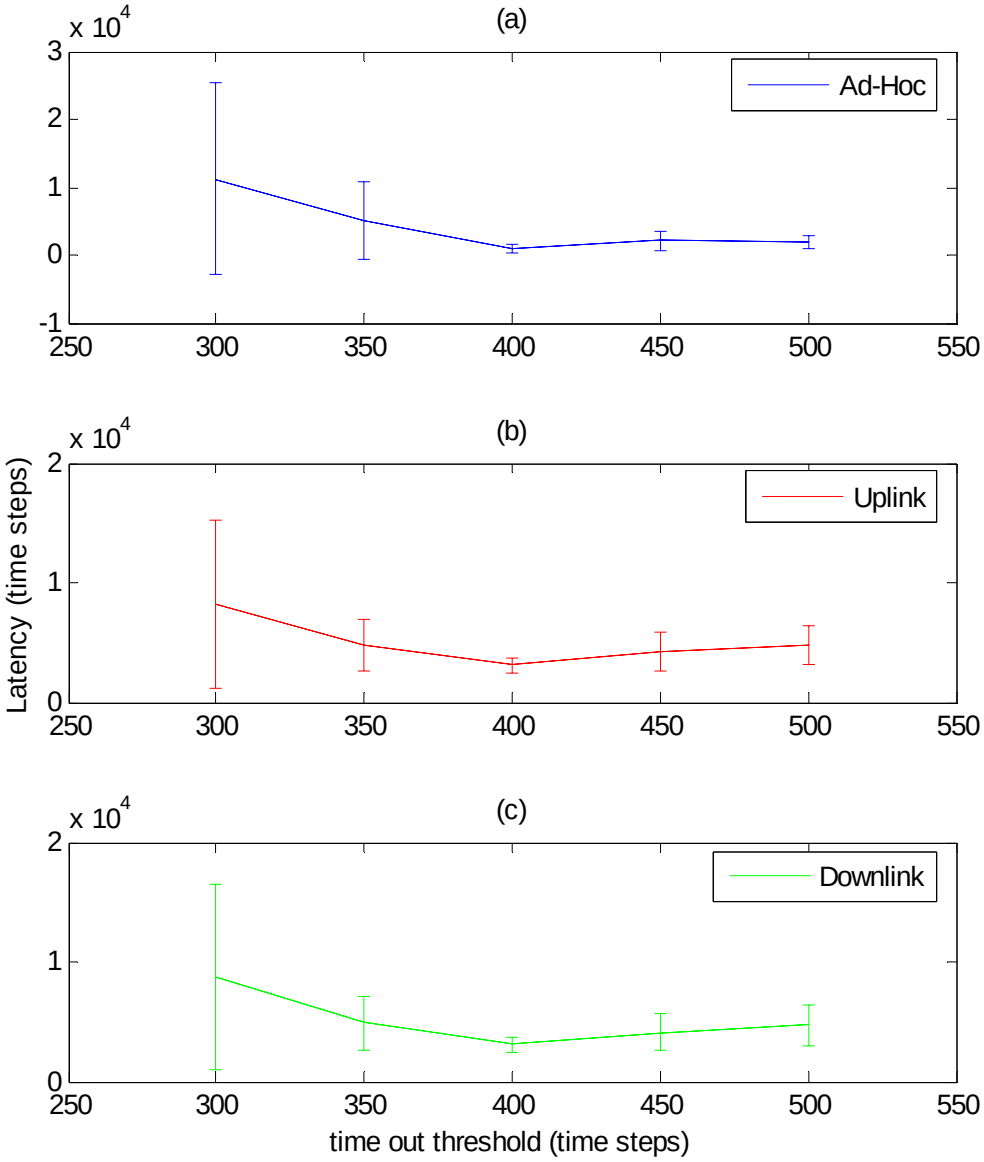

Figure 51. Comparison of packet latency (for transfer and acknowledgement) as a function of ad-hoc packet time-out threshold for (a) ad-hoc packets, (b) uplink packets, and (c) downlink packets (mean and 95% confidence interval).

The results show clear and consistent improvement in both packet latency and network throughput as the threshold is increased from 300 to 400 time steps. These improvements appear in both the sample mean and sample variance. As the threshold continues to increase towards 500 time steps, the performance improvements do not continue. The packet latency is similar but slightly higher. The network throughput is slightly lower with the exception of the COTM channel throughput at a threshold of 450 time steps. The mean performance for this threshold setting was slightly higher, but so was the variance. Based on these results, the Model 2 time-out thresholds were set at 0.502 seconds for COTM packets and 0.05 seconds for ad-hoc packets for the remainder of the performance investigations.

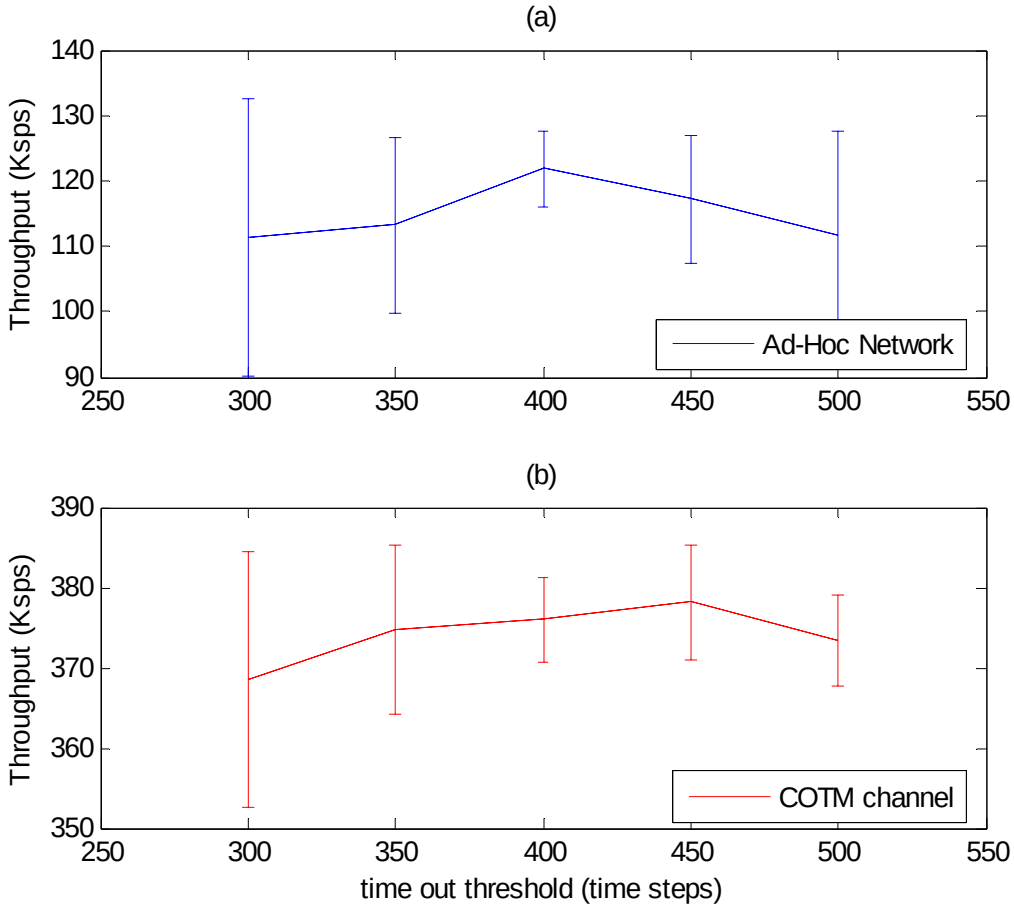

Figure 52. Comparison of channel throughput as a function of ad-hoc packet time-out threshold for (a) ad-hoc network and (b) COTM channel, uplink and downlink (mean and 95% confidence interval).

3. Summary of Time-out Threshold Parametric Study

It is important to note that only a subset of this parametric study is presented here. As part of this investigation, a short study of time-out threshold versus packet generation rate was performed. An important result of that larger investigation was the relationship of optimal time-out thresholds to packet generation rates. It has already been mentioned that setting an optimal time-out threshold is a fine balance between regenerating lost packets as soon as possible and not regenerating packets that are simply delayed by congestion. Packet generation rates are a big factor in network congestion. As more packets are being generated, the queues are longer and the congestion delay gets longer. This, in turn, shifts the optimal time-out balance. It is clear that the optimal watchdog timer function is dependent on packet generation rates. As mentioned earlier, this is not the central investigation of this research, and it was decided to not pursue an adaptive watchdog timer function (adaptive to congestion). Instead, we loosely optimized the threshold for a medium-high packet generation rate of 62.5 pps and held it fixed for all samples from that Model. A summary of those settings are found in Table 20.

Table 20. Summary of watchdog timer settings for remainder of Chapter IV.

(time step = 0.000128 sec)	Ad-hoc packet time-out	COTM packet time-out
Model 1 and 1A	200 time steps, 0.0256 sec	3202 time steps, 0.41 sec
Model 2	400 time steps, 0.0512 sec	3952 time steps, 0.506 sec

C. APPLICATION OF THE CENTRAL LIMIT THEOREM

As mentioned in Section A, simulation runs were limited to 400,000 time steps primarily due to runtime and computer memory limitations. With a TDMA slot duration set at 128 symbols and symbol durations set at 1 μs, this equates to simulating 51.2 seconds of actual network performance. For a stochastic process like network performance, 51.2 seconds is not an appropriate process sample to estimate long-term performance. A better approach is to generate multiple independent process samples and apply the Central Limit Theorem. According to [62], for processes that are not highly

skewed, 10 to 30 trials are usually necessary to invoke the central limit theorem. The presented results are based on the mid-point of this range: 20 independent and identically distributed samples.

By iterating the simulation with random initial parameters and random input vectors for COTM channel state, we can model the results as Gaussian. The results presented in this chapter represent the mean and variance of 20 independent simulations each 400,000 steps in duration. Summary results found in Figures 57, 58 and 60 through 64 are presented as sample means with their 95% confidence level. The confidence level is not the same as sample variance (second moment), but they are related through the relationship for Gaussian random variables $\sigma/\sqrt{n}$, where σ is the standard deviation and n is the number of process samples. The 95% confidence interval represents a range that the sample mean will fall within 95% of the time.

D. "BOSTON" RESULTS

The first urban environment investigated is loosely based on the empirical data collected by MIT/LL and discussed in Chapter II. As discussed in Chapter III, the simulations actually rely on a Markovian visibility chain to represent the urban blockage phenomenon. For this portion of the investigation, we have set the state transition parameters a_{ii} to approximate those seen in the empirical data (see Table 19). Therefore, we shall refer to the simulated city as Boston. Absent additional data from any other cities, we propose that the Boston visibility model represents a low-to-moderately dense urban environment. Using percent of travel distance spent in the fade state as a reasonably straightforward metric, the Boston empirical data has a density metric of 44% (31% in the temporal domain). It is important to use the spatial perspective for this kind of comparison to remove the dependence of vehicle velocity.

1. Model 1 Results

In Model 1, the COTM channel and the local area ad-hoc network are separate and independent for purposes of packet routing. If a mobile generates a packet with a

local destination, it transmits that packet on the ad-hoc network. Similarly, if a mobile generates a packet with a non-local destination, it transmits that packet on the COTM link which is the mobile's connection to the GIG.

The primary purpose of Model 1 is to establish a baseline for comparison, but Model 1 can also provide us some insight into the basic network performance and its dependencies. To present and discuss baseline network performance, we use both specific examples and a general investigation into network dependencies. The specific examples focus on buffer utilization in order to understand how the model reacts to network traffic. The general investigation into performance dependencies is important because the simulation represents a complex blend of stochastic processes (i.e., vehicle motion, COTM fade, packet generation bursts, symbol errors, etc.) and a single sample cannot confidently identify those dependencies.

a. Example 1: Nominal Performance

Example 1 presents a single mobile's buffer utilization from Iteration 3 of 20 for the combined mobile packet rate of 62.5 pps (0.008 ppts). This example is offered as a representation of nominal performance, where nominal is defined to mean that the network is not overstressed or congested and performance approximates the long-term process mean. This specific example was selected for comparison because it included both a long duration fade (fade 1) and several short duration fades (fades 2-6), which highlight the relative impacts of fade duration. For direct comparison, this example is updated in the Model 2 discussion in Section 3.

Table 21 provides some key summary statistics for this process sample. The mobile generates a total of 3115 packets for a combined packet generation rate of 0.0078 ppts, which is slightly below the expectation of 0.008 ppts. Of that 0.0078 packet generation rate, 0.00467 ppts (59.97%) are intended for the COTM uplink. This ratio nearly matches the expectation of 60%. Similarly, the satellite generates 1898 packets

for the mobile. This equates to a packet generation rate of 0.00475 ppts or 37.1 pps, and is slightly above the expectation of 0.00468 ppts. The connection ratio is slightly below the expectation of 66.6%.

For this example, the mobile experiences six COTM channel fades within 400,000 time steps (51.2 seconds). The start, finish times, and fade duration are shown in Table 22. The first and longest COTM fade lasts 75,030 time steps, or 9.6 seconds. During that time, the uplink and downlink queues build small back-logs of 697 and 665 packets, respectively.

Table 21. Example 1 packet counts and connection ratio.

Parameter	Value
COTM downlink packets generated	1898
COTM uplink packets generated	1866
Ad-hoc packets generated	1249
Connection ratio	62.6%

Table 22. Example 1 COTM fade times.

	Time (start – finish, time step)	Duration (time steps)
Fade 1	1 – 75030	75030 (9.60 sec)
Fade 2	91101 – 100799	9698 (1.24 sec)
Fade 3	210845 – 229137	18292 (2.34 sec)
Fade 4	267907 – 285432	17525 (2.24 sec)
Fade 5	342089 – 349073	6984 (0.90 sec)
Fade 6	365373 – 388666	23293 (2.98 sec)

The active uplink packet queue grows during channel fades because it uses signal strength to determine that the channel is in fade and holds packets until the LOS is clear. The downlink queue growth is slightly more complicated. The downlink does not know that the link is blocked and continues to send packets to the mobile, but it does not receive an acknowledgement from the mobile and, therefore, regenerates those lost packets when the time-out threshold has been passed. For the downlink, packets are

cycling between the active downlink queue and the pending downlink queue. As more and more packets are generated and enter this cyclic process, the downlink queue grows in size.

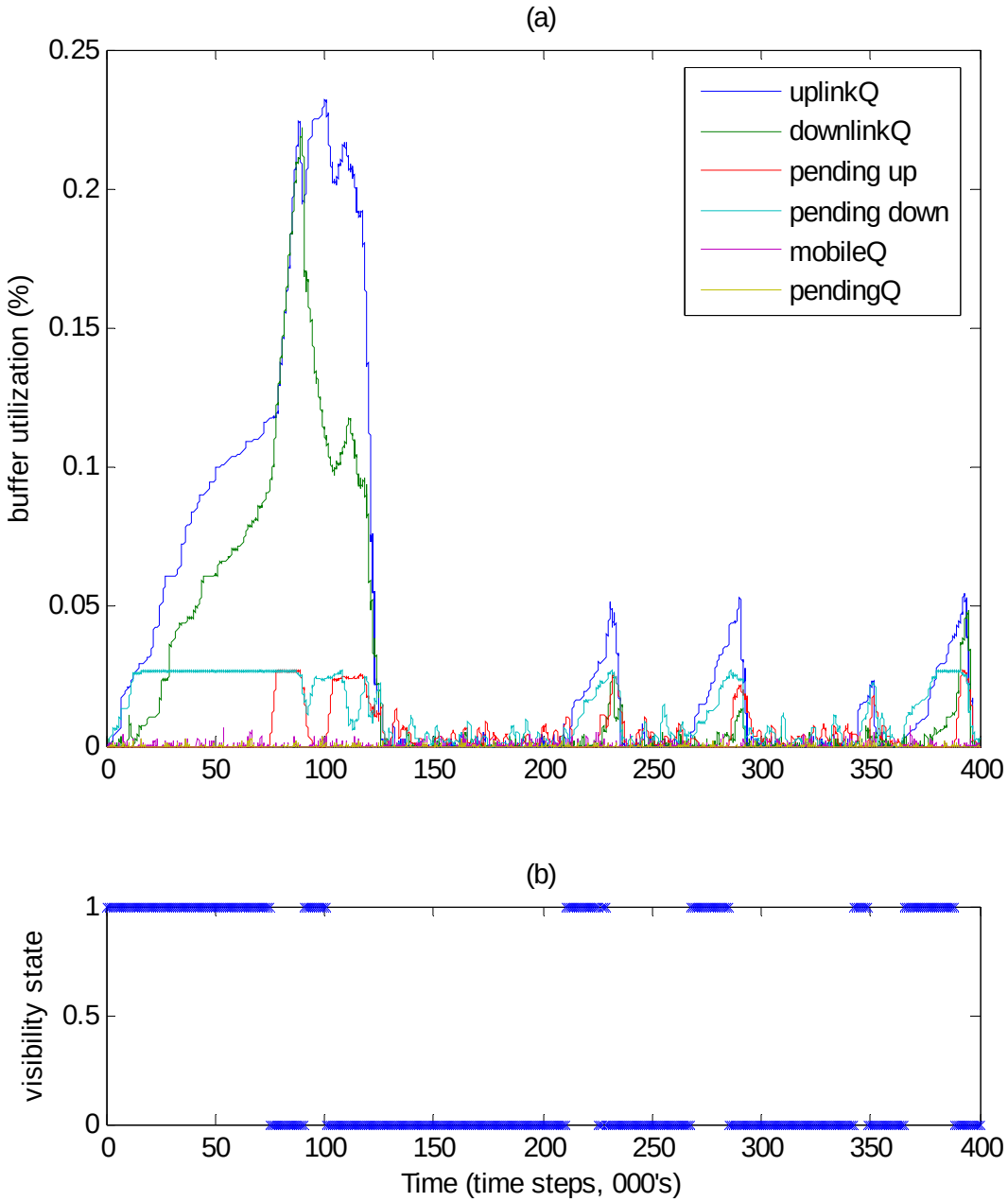

Figure 53. (a) Buffer utilization and (b) mobile visibility state for Model 1, packet rate = 0.008 ppts, Iteration 3. (visibility state: fade = 1, connection = 0).

The downlink pending queue has a different behavior that limits its growth to a ceiling determined by the packet transmittal rate and time-out threshold. At the maximum downlink transmission rate, packets enter the pending queue one every $n_{mobile} \times n_{TDMA}$ time steps, where n_{mobile} is the number of mobiles sharing the COTM channel and n_{TDMA} is the number of TDMA time slots needed to transmit an entire packet. Relevant simulation parameter settings can be found in Table 19. If the satellite is not receiving acknowledgements, packets are transferred back to the active downlink queue once the time-out threshold is reached. This combination determines that the downlink pending queue never goes above

$$\frac{\text{time - out threshold}}{n_{mobile} \times n_{TDMA}} = \frac{3952 \text{ time steps}}{5 \text{ mobiles} \times 8 \text{ time slots/packet}} = 98 \text{ packets.} \qquad (4.1)$$

In this example, the ad-hoc network is processing local area packets generated at an expected rate of 25 pps (0.0032 ppts) per mobile, well below its congestion point. Because this is an example from Model 1, the simulated system is not transferring COTM packets through the ad-hoc network. This leaves the ad-hoc network with an easy workload and the active and pending mobile queues do not grow beyond 23 and 11 packets, respectively (less than 0.01% buffer utilization).

b. Ad-hoc Network Performance Dependencies

Simulation results suggest a clear, monotonic, proportional dependence of the COTM channel performance on the packet generation rate and the fade statistics. The same relationships are not as clear for the ad-hoc network. As part of collecting and analyzing simulation results, we observed large variations in ad-hoc network performance even when model input parameters were held constant. In an effort to understand this erratic performance, we attempted to establish relationships with key driving parameters. Initial expectations focused primarily on packet generation rate and vehicle separation distance as the key dependencies. The first can increase congestion simply by filling buffers with more information to transmit. The second factor can increase congestion and impact performance by reducing link quality and causing ad-hoc

network bottle-necks. The purpose of the following investigation is to assess the relationship of *mean vehicle separation* and *packet generation rate* to ad-hoc network performance, where *mean vehicle separation* is the average vehicle separation over one simulation run of 400,000 time steps or 51.2 seconds.

The performance comparisons presented in Figures 54 through 56 provide some insight into these complex relationships and provide a means to assess the relative impacts of packet generation rate and vehicle separation distance. Together these three figures show the key ad-hoc network performance averages for 80 independent process samples, 20 for each packet rate. The results are presented as scatter plots due to the stochastic nature of the vehicle separation. The packet generation rates shown in these figures are for ad-hoc packets generated by a single mobile and the units are packets per second. From lowest-to-highest, these four packet rates move closer to the observed congestion range for the ad-hoc network which starts near 40 pps. The two primary dependency candidates, mean vehicle separation and packet generation rate, are uncorrelated. This is reflected in the 20-sample average of mean vehicle separation as compared to packet generation rates, where all four mean separations fall within a tight band less than 10 meters wide (see Table 23).

Table 23. Packet generation rate vs. mean vehicle separation

Packet generation rate (pps)	18.75	25.0	31.25	37.5
Mean separation (meters, 20 sample average)	184.4	185.7	174.7	183.4

Figure 54 shows 80 different samples of mean packet latency, 20 for each packet generation rate. Analyzing the close-up view presented in Figure 54(b), the samples show a noticeable increase in latency as the average mobile separation grows from approximately 110 meters to over 250 meters. Similarly, the packet generation rate appears to have some impact on latency as well. The 20 sample averages are shown in Table 24.

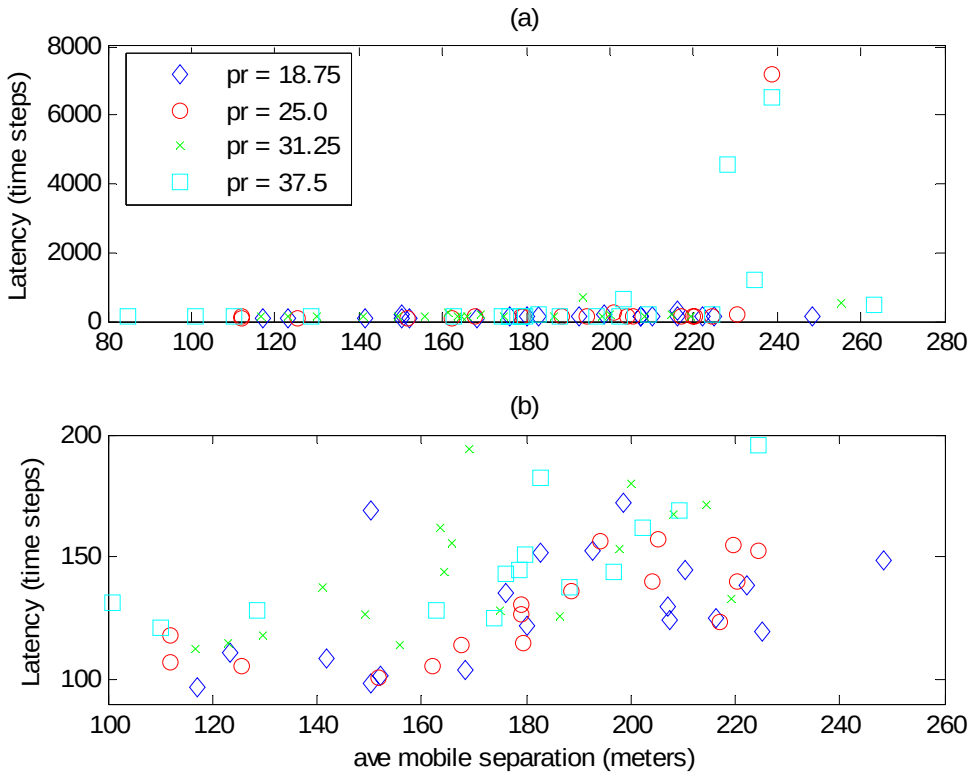

Figure 54. Mean ad-hoc packet latency as a function of average separation distance and packet generation rate; (a) all data, (b) zoom into latency range 90-200 time steps.

Table 24. Packet generation rate vs. mean packet latency

Packet generation rate (pps)	18.75	25.0	31.25	37.5
Mean latency (time steps, 20 sample average)	136.8	491.3 (138.8)	197.6	778.2

The 491.3 time step result is surprisingly high for packet rate 25.0 pps. Additional investigation shows that this result is heavily skewed by a single problematic simulation run that had an average packet latency of 7189.3 time steps. After removing this outlier, the remaining 19 process samples have a revised mean latency of 138.8 time steps and matches expectations. A focused investigation of the outlier found that the extremely poor performance was due to large vehicle separations (mean of 238 meters) and a difficult node topography. This is discussed more later in this section.

As the packet generation rate approaches 40 pps, ad-hoc performance becomes more prone to congestion and degraded performance as reflected in the outliers for 37.5 pps. The combination of the higher packet generation rate and the above average mobile separation can lead to large queues and noticeably longer wait times. Clearly the combination is an important factor. When the packet generation rate drops to 25 pps, even an average separation of 250 meters only increased packet latency to 150 time steps.

Figure 55 is similar to Figure 54 except that is shows network throughput in ksps. Here again we see a small but noticeable dependence on average separation distance, which is reflected in a slight downward slope. Even when comparing the best performance to the worst, we find less than 20% drop in throughput as the mean separation distance is more than doubled.

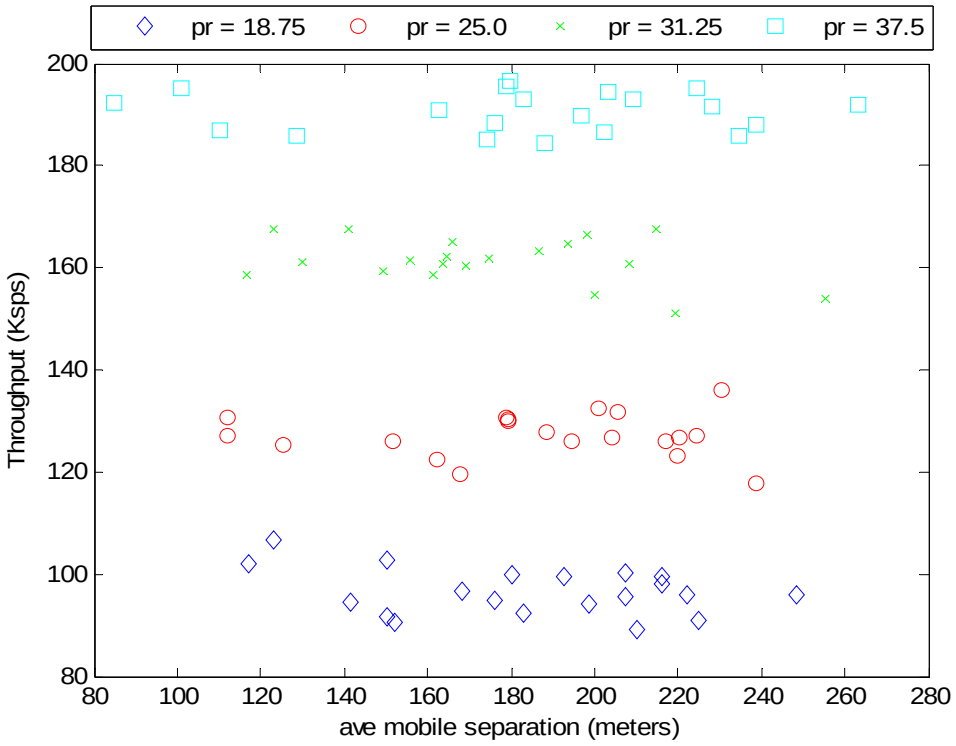

Figure 55. Ad-hoc network throughput as a function of average separation distance and packet generation rate.

Figure 56 is the final assessment of network performance as a function of packet generation rate and mean vehicle separation. This figure compares the average number of transmit attempts required to send a packet and receive an acknowledgement.

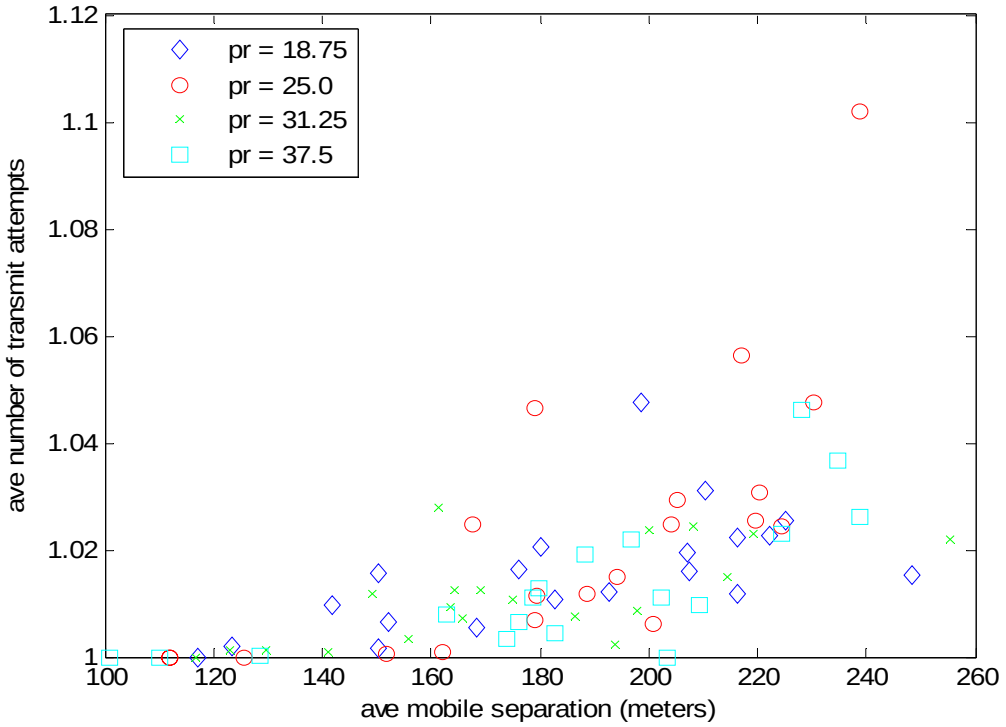

Figure 56. Average number of transmit attempts as a function of average separation distance and packet generation rate.

The results of this investigation show some general trends and relationships between network performance, mean separation distance and packet generation rate, but the highly scattered nature of the performance results show that network performance is not necessarily a proportional or monotonic dependence on packet generation rate or mean separation distance. The figures show several data points that are difficult to explain based on the first order parameters of packet generation rate and mean separation. One example is the 170 time-step packet latency achieved at a packet generation rate of 18.75 pps and a mean separation of only 148 meters. This is the

second worst mean latency for 18.75 pps but it occurs with a mean separation that is well below average. This is only one example amongst many where performance improves despite increases in packet rate and/or vehicle separation.

Additional investigation into the specifics of these results showed that mean separation distance was not as important as the specific node topography. In each of the cases where performance was surprisingly poor, the node topography was such that three or four of the mobiles were spread far apart and one mobile was in the center. In this situation, the center mobile becomes a conduit for all local area traffic and has more packets to process than usual, leading to increased queue wait times. The results of this investigation show that while mean separation distance and packet generation rate provide acceptable first-order estimates of ad-hoc network performance, node topography is often a larger influence.

2. Model 1A Results

Model 1A is nearly identical to Model 1 with the exception of a minor change to the downlink routing protocol. The modification involves putting the downlink transmission into a wait state when the satellite can no longer detect the mobile's uplink carrier. The satellite keeps the downlink in this wait state until it detects the mobile's uplink carrier or its associated timer has counted down. Figures 57 and 58 show a statistical summary comparing the Model 1A performance with that of Model 1. The figures show mean performance with the 95% confidence interval for 20 independent iterations for Model 1A but show only mean performance for Model 1. All of the results are based on the same visibility and packet generation vectors. The comparison includes a variation of the Model 1A wait-state timer duration from 0.5 to 3.0 seconds. This range was chosen because it brackets the average fade duration shown in Table 4.

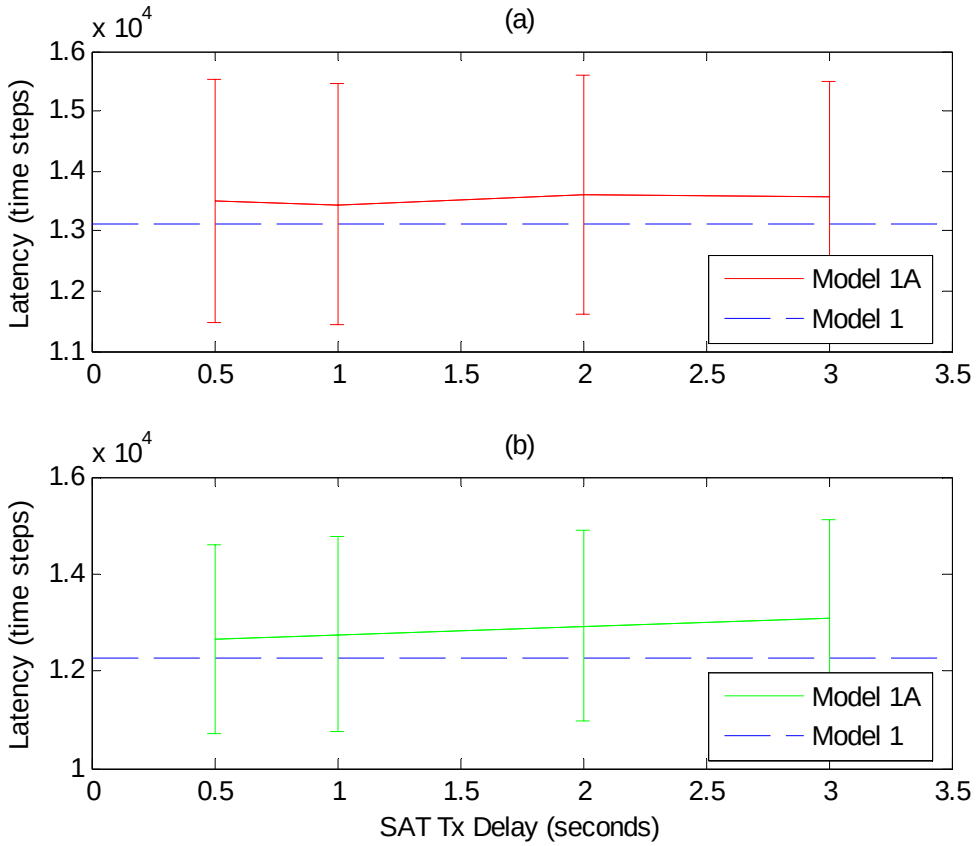

Figure 57. (a) Uplink and (b) downlink packet latency vs. wait state duration as compared to Model 1 performance (mean and 95% confidence interval, COTM packet generation rate = 0.0048 ppts / 37.5 pps).

Figures 57 and 58 show that the Model 1A wait-state protocol performed worse than the baseline in both packet latency and channel throughput for all four wait-state durations. Additional investigation found that Model 1 performed better simply due to the fact that it made more frequent and consistent attempts to transmit packets to the mobile. While transmitting packets to a blocked mobile may be inefficient, more frequent attempts increases the chance of successful transfer. In many cases, Model 1A avoided meaningless packet transmission to a blocked mobile that could not receive the packet (see Figures 43 and 46). The problem is that avoiding extraneous channel activity does not really matter to a channel that is operating well below the congestion point and has capacity to spare. On the other hand, Model 1A still misses valuable transfer

opportunities when $t_f < 2t_d + t_w$ (see Figures 44 and 45). The bottom line is that when the channel is operating nominally (not congested), the inefficiency of extraneous packet transfer is not nearly as important as the inefficiency of missing valuable connection windows. Due to a lack of positive impact, Model 1A will not be explored further.

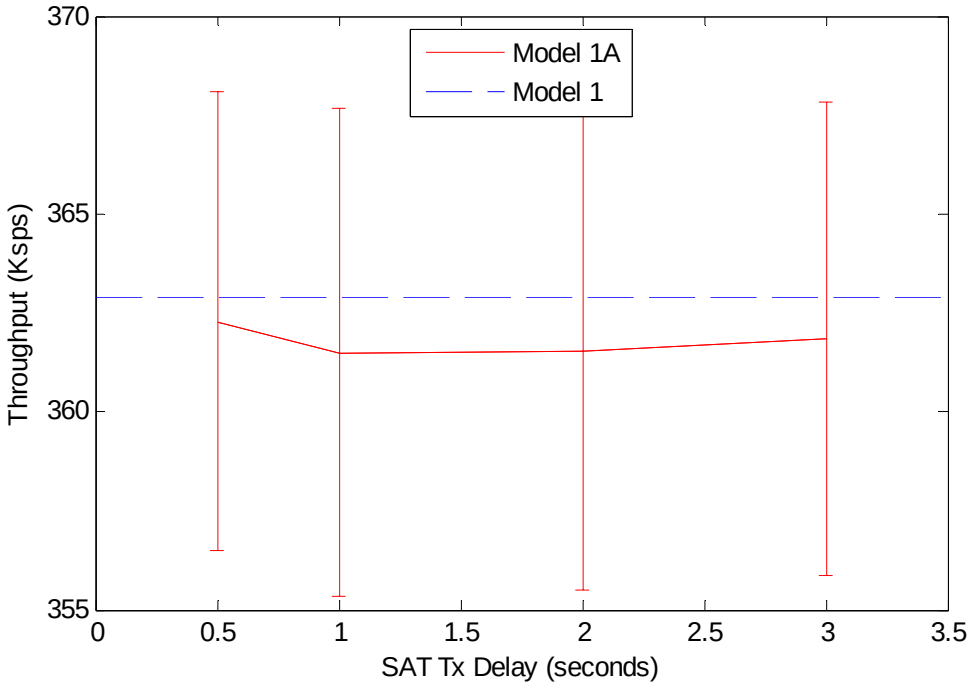

Figure 58. COTM channel throughput vs. wait state duration as compared to Model 1 performance (mean and 95% confidence interval, COTM packet generation rate = 0.0048 ppts / 37.5 pps).

3. Model 2 Results

For comparison of Model 2 with Model 1 performance, we use both specific examples and general statistics. The specific example is carried over from the Model 1 discussion in Section 1 and focuses on buffer utilization. The focused example is an important approach for understanding how each model responds to a specific situation. The summary performance statistics are also important because the simulation represents

a complex blend of stochastic processes (i.e., vehicle motion, COTM fade, packet generation bursts, symbol errors, etc.) and a single sample cannot confidently represent the long-term performance.

a. Example 1: Nominal Performance

In the Model 1 discussion of Section 1, Figure 53 presents a single mobile's buffer utilization Iteration 3 of 20 for the combined mobile packet generation rate of 62.5 pps. Figure 59 shows an update of the same example when operating under the Neighbor protocol. It is immediately apparent that performance has changed dramatically. The supporting visibility vector shows that the COTM channel experiences the same fades as before, but they no longer cause the uplink or downlink queues to grow in size. In fact, the uplink queue never gets above 50 active packets present at one time. Instead, those packets are transferred to the ad-hoc mobile queue which transmits them to a neighbor for surrogate transmission. Likewise, the downlink packets are received by a neighbor and transferred on the ad-hoc network.

For this example, the mobile experiences six COTM channel fades within 400,000 time steps (51.2 seconds). The start time, finish time, and duration are summarized in Table 22. The first and longest COTM fade lasts for approximately 75,000 time steps or 9.6 seconds. Figure 53 shows that when Model 1 experiences this situation, the active uplink and downlink queues grow substantially in size, while the active ad-hoc queue shows no noticeable change in activity. Model 2 has a completely different reaction to the 9.6 second fade. Instead of monotonically growing to 697 or 665 active packets, the active uplink and downlink queues experience a relatively minor change in activity and never grow beyond 87 and 55 packets, respectively. The uplink/downlink pending queues also have a different response to the situation with more erratic behavior reflecting the fluctuating receipt of acknowledgements.

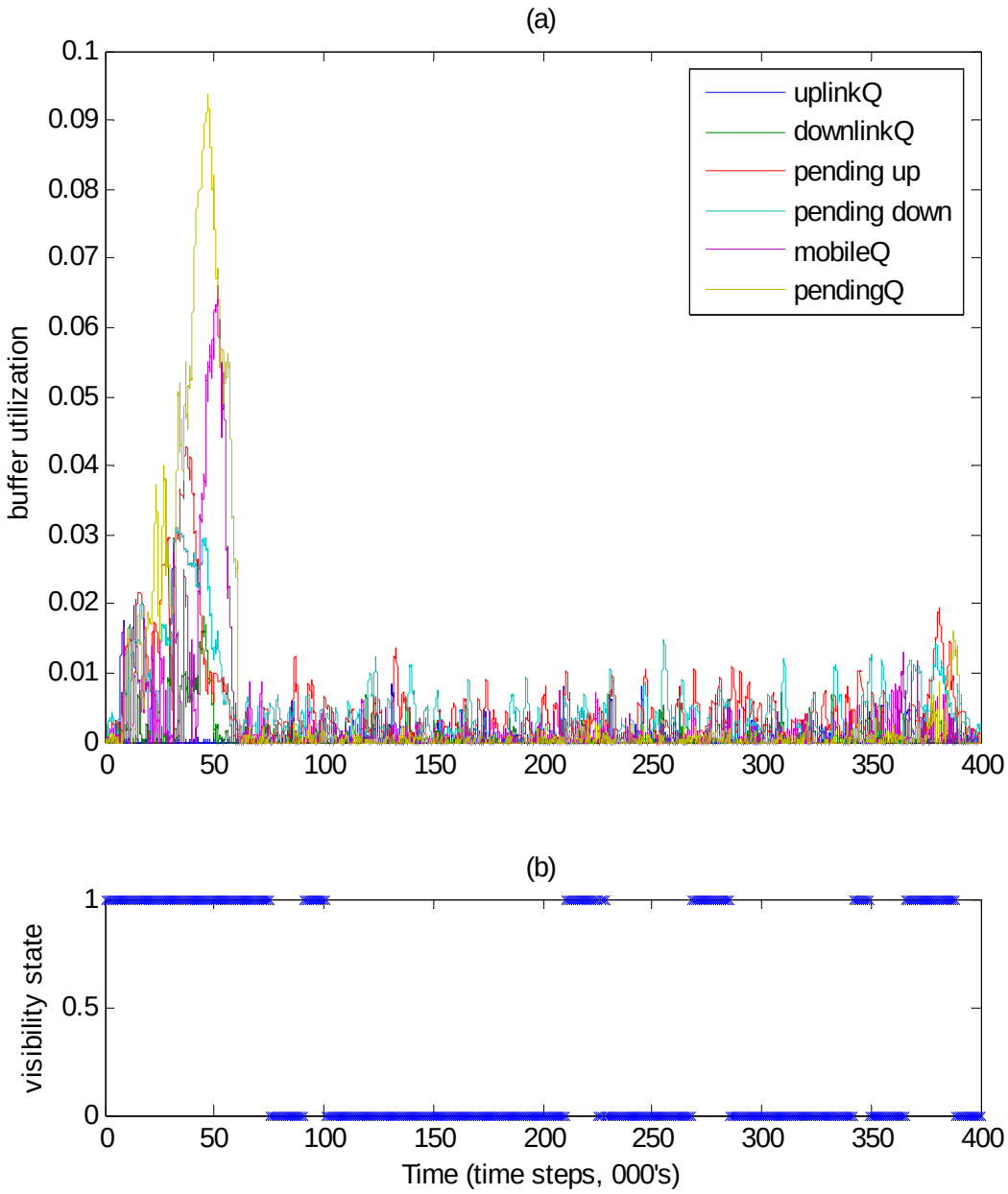

Figure 59. (a) Buffer utilization and (b) mobile visibility state for Model 2, packet rate = 0.008, Iteration 3. (visibility state: fade = 1, connection = 0).

The relatively subdued response of the four COTM queues is a direct result of the Neighbor protocol. The active COTM packets are no longer waiting in the mobile's uplink queue or being lost on a blocked downlink. Instead, uplink packets are being transferred to a neighbor and being sent to the satellite. Downlink packets are

being collected by a neighbor and transferred to the destination mobile via the ad-hoc network. While these changes certainly provide the COTM buffers with milder fade impacts and reduced packet service times, the changes are not without a cost. In Figure 53, the ad-hoc queues are handling the local area packet traffic with ease, rarely building a packet queue of even 15 active packets. In Model 2's response, the 9.6 second fade has a noticeable impact on the mobile's active and pending ad-hoc buffers. Instead of hovering below 15 active packets, the active and pending ad-hoc queues climb and drop erratically with maximums of 198 and 281 packets, respectively. These larger queues reflect an increase in packet service time for the ad-hoc network which is discussed further in the summary statistics.

b. Comparison of Summary Statistics

While comparison of specific examples is helpful for understanding the details and mechanics of the two models, the overall performance comparison needs to be more broadly based on statistical results for these multi-dimensional stochastic processes. Figures 60 through 62 compare the network performance of both models using the same visibility, packet generation, and vehicle location vectors. The only remaining stochastic activity that is different between the two implementations is the computation of individual symbol errors (Bernoulli trial). The results are presented as functions of combined packet generation rate, ad-hoc and COTM, for an individual mobile. As discussed in Section B, packet generation is weighted 60% COTM channel and 40% local destination.

Figure 60 compares packet latency between the two models for ad-hoc, uplink and downlink packets. The uplink and downlink packets are kept separate to investigate any significant variations in performance due to the path delay. The results show significant reductions in COTM packet latency, especially for the the lower two packet rates. The average improvement is much smaller for 78.125 pps and the variance is larger. This is a result of the ad-hoc network occasionally running close to its congestion point and falling behind on packet processing. Sometimes it takes a packet

just as long to navigate the ad-hoc network as to wait for the next LOS connection. As expected, improvements in COTM latency are not without a cost. The ad-hoc packet latency is consistently higher for Model 2 than for Model 1. The exceptionally large latency and variance for 78.125 pps shift the scale to make the smaller impacts difficult to read. The Model 2 results show an increase in ad-hoc packet latency of 123%, 112% and 9050%, for 46.875, 62.5 and 78.125 pps, respectively. Even though 123% and 112% seem large as percentages, the latencies are still very small (a maximum of 0.22 seconds) and are not unacceptable to local area network performance. It is certainly a reasonable trade for the significant improvements in COTM packet latency.

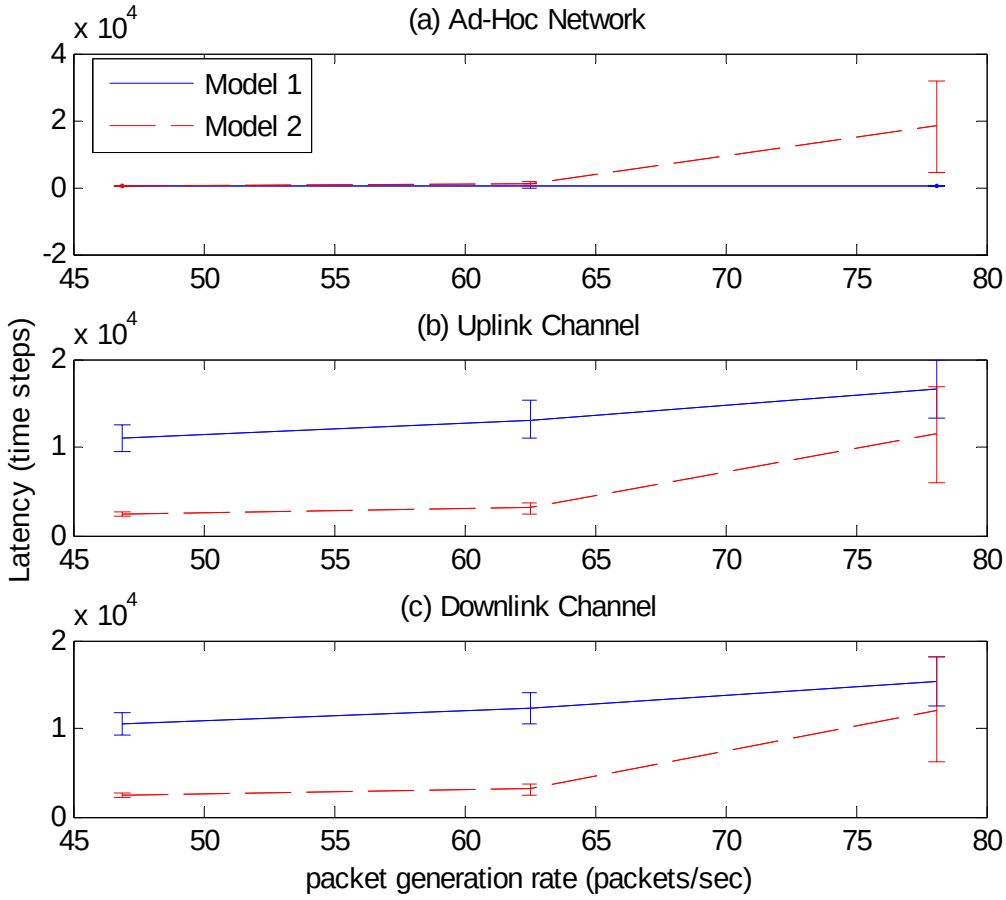

Figure 60. Comparison of packet latency between Models 1 and 2 for (a) ad-hoc network, (b) uplink channel, and (c) downlink channel (mean and 95% confidence interval).

Ignoring the congested performance at the highest packet rate, the minimum COTM latency savings is a 76% reduction in packet transmittal and acknowledgement. This represents a drop from 1.17 to 0.28 seconds.

The next area of performance comparison is network throughput. Figure 61 shows the statistical performance for each model and packet rate. The COTM channel and ad-hoc network are shown separately. This performance parameter experienced similar changes to those seen with packet latency. For the lower two packet rates, COTM channel throughput improved 2.6 - 3.7%, while the ad-hoc network throughput dropped 0.0 - 7.2%. The improvements are smaller for throughput than latency because throughput is constrained by the number of packets generated. Using the number of packets generated as an ideal, we see that the Model 2 results are within 0.1% of ideal. The highest packet rate did not provide any improvements in performance but instead struggled with congestion problems that hampered both networks.

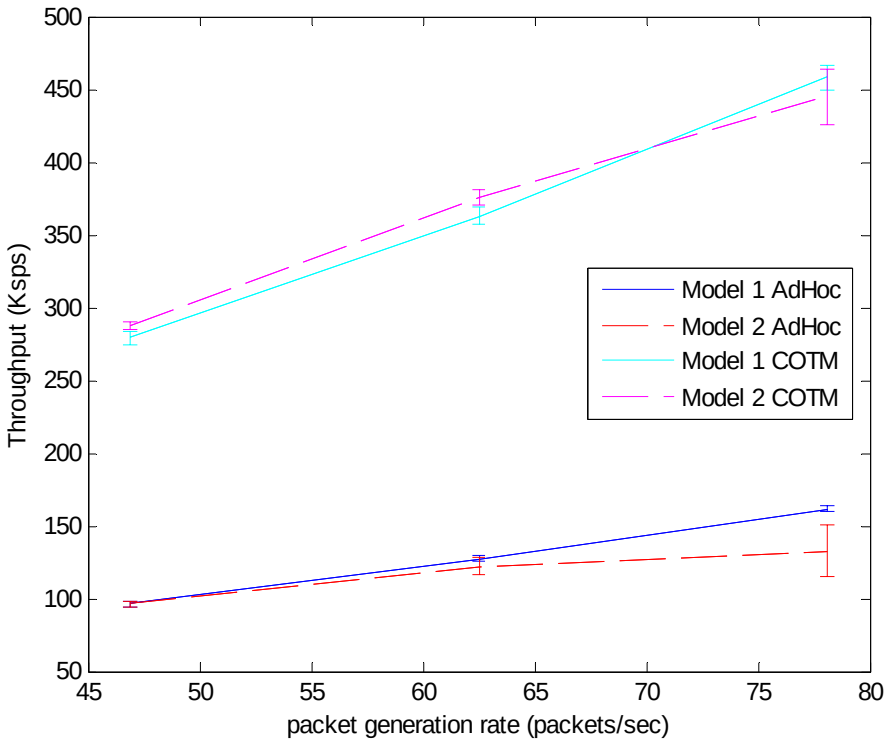

Figure 61. Comparison of throughput between Models 1 and 2 for ad-hoc network and combined COTM channel (mean and 95% confidence interval).

Table 25. Model 1 vs. Model 2: Relative mean performance (Boston)[1].

Total packet rate (pps)	COTM latency	MANET latency	COTM throughput	MANET throughput
46.875	−78%	+123%	+3%	0%
62. 5	−76%	+112%	+4%	−4%
78.125	−26%	+9050%	+3%	−17%

[1] Note: green text = improvement, red text = degradation.

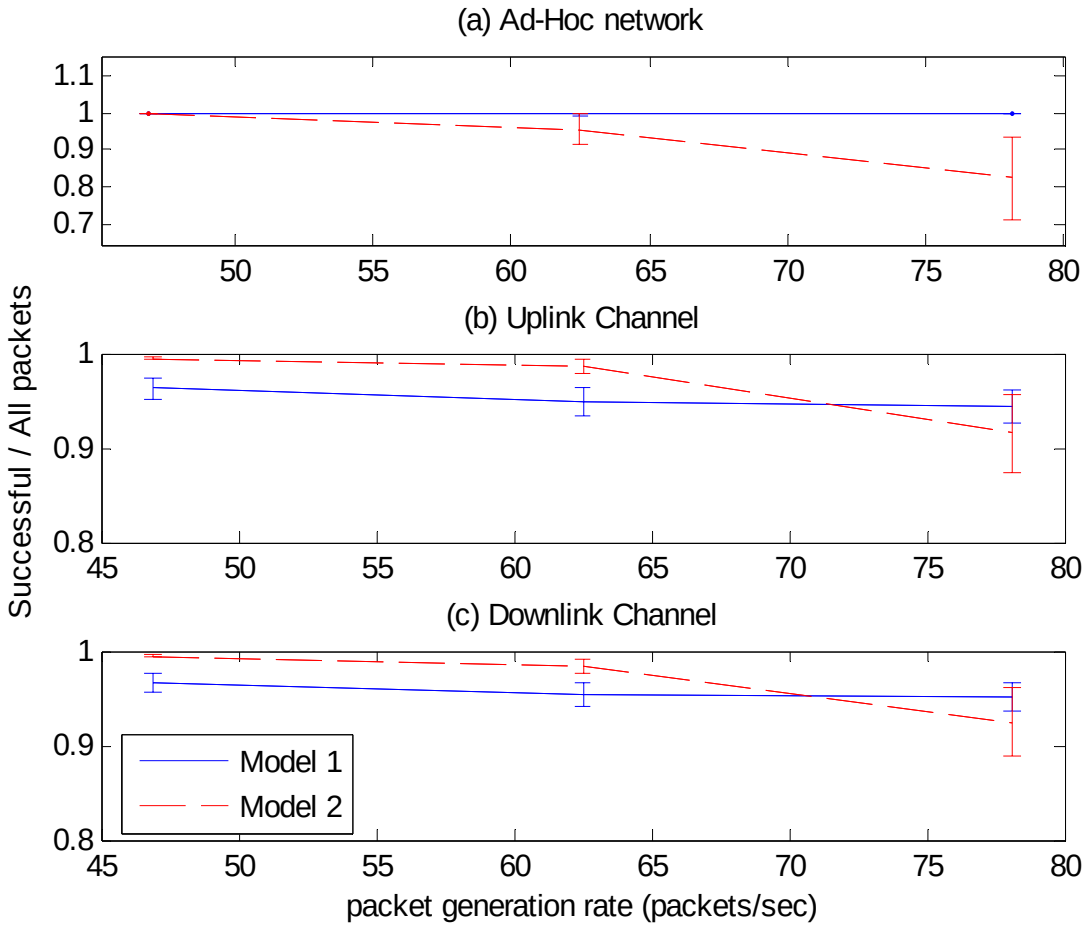

Figure 62. Comparison of packet success ratios between Models 1 and 2 for (a) ad-hoc network, (b) uplink channel, and (c) downlink channel (mean and 95% confidence interval).

E. "NEW YORK" RESULTS

One of the most significant factors impacting network performance is the *density* of the urban environment, where density refers to the long term ratio of time spent in connection to time spent in fade. In Section D, we presented results for a low-to-moderately dense urban environment that was intended to represent Boston, MA. It is also important to understand how the baseline and Neighbor protocols perform in a denser urban environment. To represent this increased urban density, we reversed the Markov visibility chain transition parameters a_{ii} as shown in Table 19. Where Boston had an average connection duration of 4.2 seconds and an average fade duration of 2.0 seconds, the New York model averages 4.2 second fades and 2.0 second connections. Absent any supporting empirical data, we suggest that these visibility statistics are above average and represent a dense urban environment similar to New York City.

It is important to note that the packet generation rates analyzed for Boston were too aggressive for this type of urban environment. The simulations showed unavoidable network congestion and overflowing buffers with a significant number of packet overwrites. For the New York analysis, we analyzed packet generation rates of 7.8125, 15.625, 31.25 and 46.875 pps. These packet generation rates convert to 0.001, 0.002, 0.004 and 0.006 packets per simulation time step, respectively.

The performance results for round-trip packet latency and throughput are shown in Figures 63 and 64, respectively. Table 26 provides a numerical version for easier reading. While in general the results follow the same trends as seen in the Boston analysis, the New York results are even more impressive for the Neighbor protocol.

As the mean fade duration grows, the Neighbor protocol shifts more and more traffic to the ad-hoc network, and the ad-hoc network becomes congested sooner. At the highest combined packet generation rate of 46.875 pps, 18.75 pps have local destinations and are transported on the ad-hoc network in either model. In Model 2, the mobiles are blocked approximately 66.6% of the time, so we can estimate that at least 18.75 of the 28.125 COTM packets being generated each second are being shifted onto the ad-hoc

network as well. This does not include any packets that are regenerated during a COTM fade or waiting in the uplink queue at the time of fade entry. The combination of 18.75 ad-hoc packets and 18.75 COTM packets per second puts the ad-hoc network traffic at a minimum of 37.5 pps per mobile, which is approaching the congestion range.

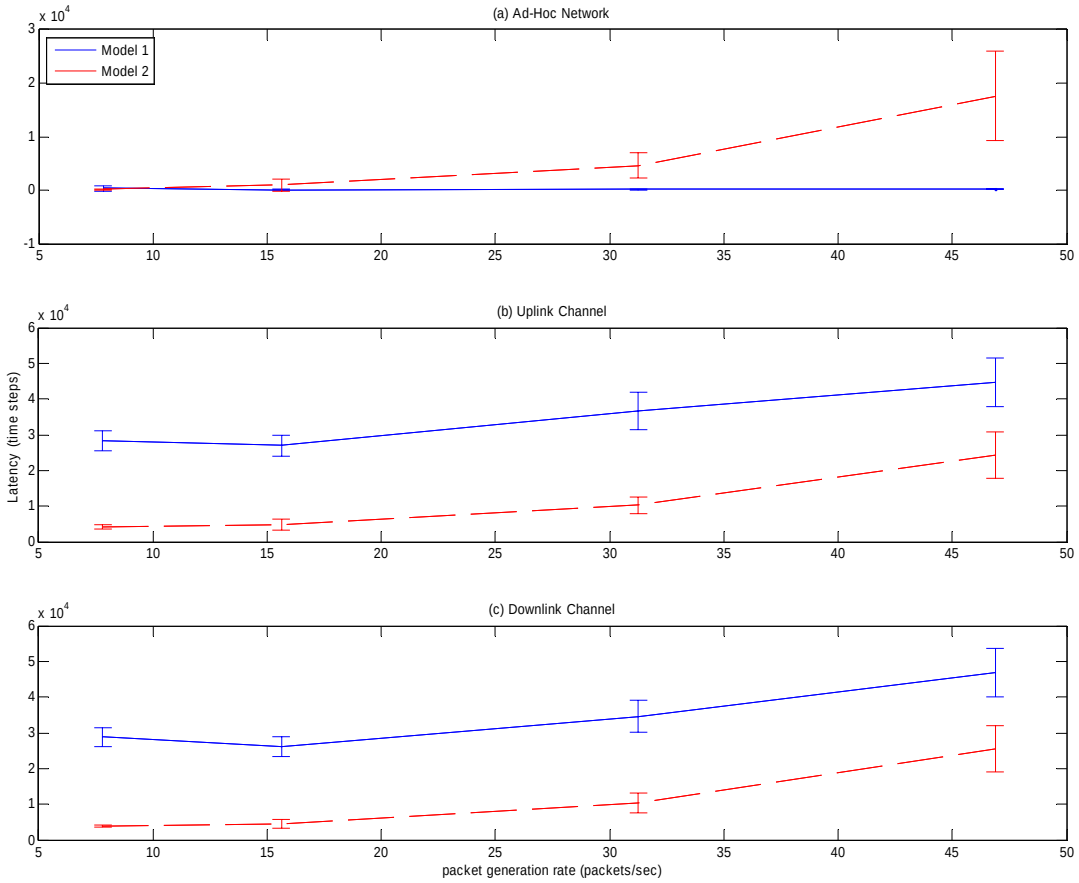

Figure 63. Comparison of packet latency between Models 1 and 2 for (a) ad-hoc network, (b) uplink channel, and (c) downlink channel (mean and 95% confidence interval).

Table 26. Model 1 vs. Model 2: Relative mean performance (New York City)[1].

Total packet rate (pps)	COTM latency	MANET latency	COTM throughput	MANET throughput
7.8125	−86%	−68%	+8%	0%
15.625	−82%	+845%	+9%	−2%
31.25	−71%	+4148%	+7%	−14%
46.875	−46%	+12823%	+2%	−37%

[1] Note: green text = improvement, red text = degradation

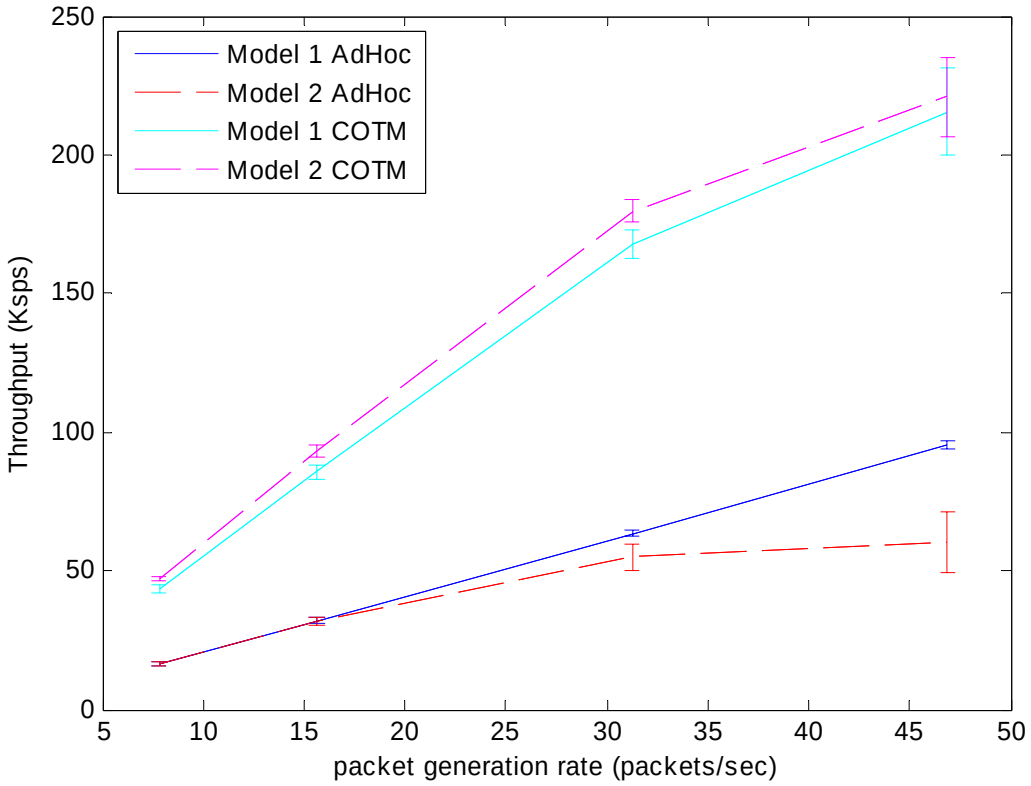

Figure 64. Comparison of throughput between Models 1 and 2 for ad-hoc network and combined COTM channel (mean and 95% confidence interval).

F. CHAPTER IV SUMMARY

The Neighbor protocol clearly and consistently provides substantial improvements in urban COTM channel performance up until the point that the ad-hoc channel becomes congested. For the specific network implementation investigated, the

MANET congestion starts to occur with increasing frequency as the packet generation rates exceed 35 to 40 kbps depending on node topography. These packets can be internally generated with local destinations or they can be transients rerouted by the Neighbor protocol. In those situations that the ad-hoc network has remaining throughput capacity available, the Neighbor protocol brings COTM packet latency within acceptable limits and mitigates the problematic impacts of long-term deep signal fades. The Neighbor protocol achieves success at reaching the second research goal of mitigating the impacts of the urban fade on network performance. The next chapter summarizes the conclusions of this dissertation and suggests ideas for follow-on research in the area of modeling urban COTM channels.

V. CONCLUSION AND RECOMMENDATIONS FOR FUTURE WORK

A. CONCLUSION

Robust and rapid communication has always been important to the success of military activities. This importance and recent technological advancements are driving the military to expend a significant amount of resources on developing an improved communications infrastructure for the next decade. The DoD's broad goal of robust and rapid communication includes a specific interest in maintaining real-time, high data rate communications with the individual vehicle or soldier while they are in motion. To achieve this facet of a future communications architecture, the military has already invested a significant amount of effort to understand and exploit the land mobile satellite channel. The military's initial research into the LMSC found the urban environment to be the most challenging due to frequent long duration signal fades caused by urban structures. The research presented in this dissertation is focused on extending our understanding and improving our exploitation of the LMSC. These goals are accomplished by building an improved channel model and developing mitigation strategies that improve channel throughput and latency.

The research presented in this dissertation includes two original and valuable contributions to the field of land mobile satellite channel modeling. The first contribution is an extension of previous efforts to model the urban communications-on-the-move channel using a hidden Markov model. By increasing model complexity and replacing the discrete transmission HMM with a continuous transmission HMM, the model provides a better match to test data. As a secondary benefit, the improved model also accurately models urban COTM signal strength as a function of time. Combining this research with previous efforts, we propose a general COTM (LMSC) channel model that has three intrastate HMMs embedded within a larger interstate HMM, where the states

are the open, rural, and urban environments (see Figure 65). Based on this concept, the urban continuous transmission COTM HMM becomes an embedded output transmission function inside the urban state.

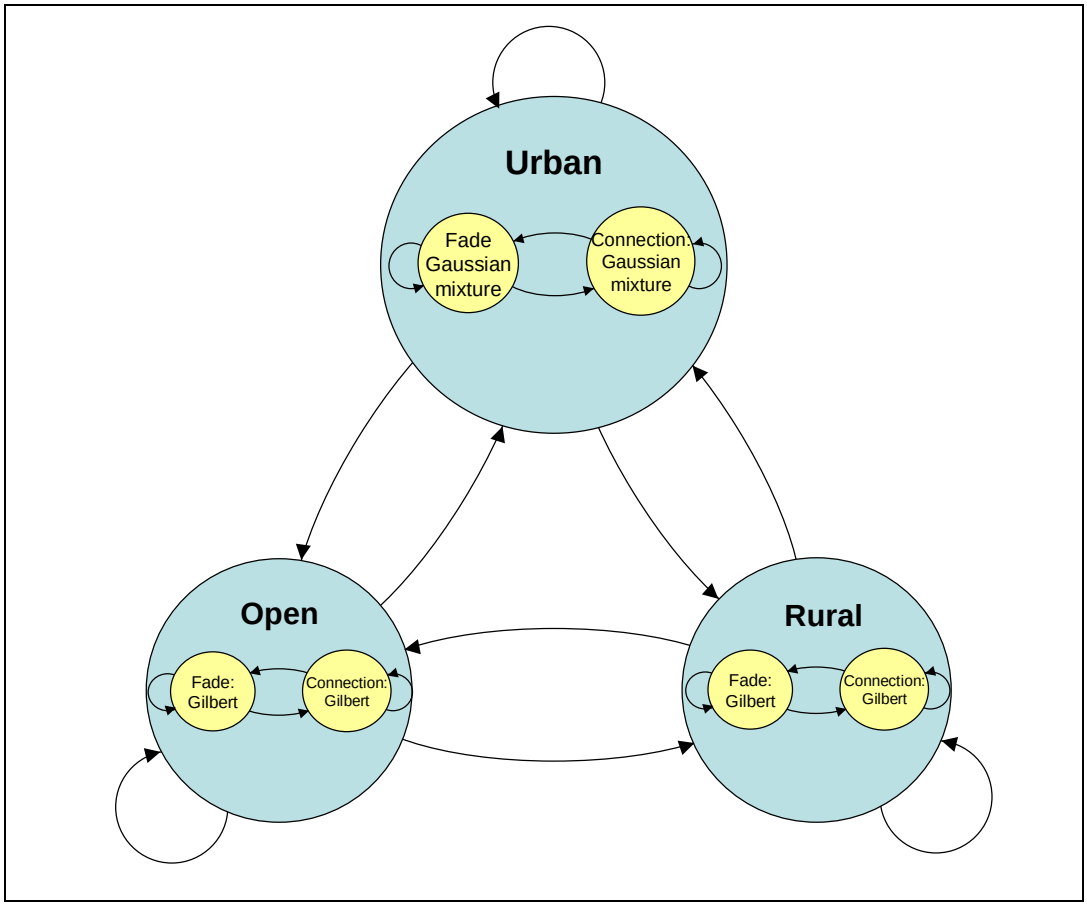

Figure 65. General COTM channel model for all environments

The second important contribution of this research effort is a time-based simulation that implements the Neighbor protocol as a mitigation for the urban environment's performance impacts. The Neighbor protocol mitigates the impacts of an urban environment by exploiting additional paths through a parallel mobile ad-hoc network and circumventing signal fades. Results show that the Neighbor protocol clearly and consistently provides substantial improvements in urban COTM channel performance up until the point that the ad-hoc channel becomes congested. The

Neighbor protocol is especially effective for TDMA-based SATCOM channel designs because neighbors are not required to sacrifice their own uplink or downlink activity.

The computer-based simulation is also a valuable tool in its own right. It provides a low cost means for estimating the performance of a complex network scenario with multiple nodes moving in an urban environment. The simulation offers a means to explore both COTM channel and ad-hoc network performance while varying key input parameters which can provide insight into additional performance dependencies. With minor modifications, the simulation could also be used to explore new mitigation strategies.

Fade prediction is another possible approach to mitigating the urban COTM channel fade impact. Efforts to successfully predict channel fades with adaptive linear filters were hampered by low temporal resolution in the empirical test data. Appendix A provides a discussion of this approach, the performance to date and offers suggestions for future work on fade prediction.

B. RECOMMENDATIONS FOR FUTURE WORK

Chapters II – V contain several suggestions for future work on these topics. A consolidated list of the most important suggestions is repeated in this section.

1. Empirical Data Collection

As mentioned in Chapter II, this research area could be improved with more diverse and higher resolution empirical data that includes a range of cities and multiple routes through those cities.

The increased diversity provided by multiple routes and multiple cities would be beneficial in two ways. First, the TCA system design could more accurately develop performance expectations and critique mitigation strategies. Second, this type of empirical data set would be useful for expanding or improving the effort to develop a

stochastic (e.g., Markov) model. Chapter IV presents results for a fictitious city labeled "New York", but we do not really know what the appropriate model parameters are for the real New York city.

As mentioned in Chapter V, increased resolution is crucial to improving the performance of a linear prediction filter. As the temporal resolution of the data improves, the ULMSC fade transitions will become less abrupt and provide some concrete performance data on how much of an early warning diffracted energy would provide.

2. Network Modeling and Simulation

a. Inclusion of Packet Execution Processes

As discussed in [5], packet execution processes (i.e., mux/demux, interleaving, encoding, and encryption), are significant in today's communications systems. Modifying the simulation to include delays for supporting packet execution processes will increase the accuracy of the performance estimates.

b. Network Performance against Jamming

The GIG and TCA are military systems that are expected to face jamming attempts by hostile forces. It is important to understand the performance of the COTM link, the MANET and the Neighbor protocol in this situation. If the simulation were modified to include hostile interference, it could be used to investigate network response and performance to local area jamming.

c. Adaptive Links

Since TSAT is planning to implement an adaptive link margin protocol, model improvements to include an adaptive link margin protocol could be useful. The adaptive protocol might manipulate code rate, signal power, data rate or some combination of all these parameters. This simulation improvement would be most appropriate when coupled with the narrow-band interference or hostile jamming.

d. Full-duplex Ad-hoc Communication Systems

The results presented in Chapter IV show that the success of the Neighbor protocol is influenced by the capacity of the supporting MANET. The current simulation allows for changes to instantaneous data rates, time-out thresholds, and FEC performance. Modification of the simulation to include a full-duplex MANET would provide additional performance estimates and support resource decisions made by the system design team.

e. Perfect vs .Realistic Wide Information

Currently, Model 2 assumes perfect side information regarding a neighboring mobile's COTM link state (fade/connection) and buffer utilization. Removal of these assumptions and modeling a delayed administrative network responsibility to spread this information would provide an estimate of the impact of the additional overhead. More importantly, this modification could also identify a drop in network performance due to the protocol making decisions with old information.

f. Modifications to the Ad-Hoc Forwarding/Routing Protocol

As mentioned in Chapter III, the simulation implements an ad-hoc forwarding/routing protocol of low-to-moderate complexity. Many authors have explored additional modifications, complexities and improvements to ad-hoc routing protocols. Some suggestions focused on the LRR metric are presented below.

In [55, 56], the authors introduce the idea of routing based on packet type where the examples are voice packets and data packets. Voice and data packets are merely two examples that help illustrate realistic variations in network performance requirements. While voice packets cannot tolerate much delay, they can tolerate some erasures and errors. On the other hand, data packets have more stringent error requirements but may allow for a moderate delay in the transfer. The LRR metric equation presented in Chapter III is repeated here with additional subscripts of v or d for voice or data respectively:

$$LRR_d(A,B) = \alpha_d I_d(A,B) + \beta_d W_d(B), \quad (5.1)$$

$$LRR_v(A,B) = \alpha_v I_v(A,B) + \beta_v W_v(B). \quad (5.2)$$

Each of the two resistance terms has a weighting coefficient (α, β) that allows the metric to be tailored to packet specific requirements. The *d* subscript denotes a metric specifically for data packets. For a protocol that delineates between packet types, a separate routing table needs to be maintained for each type of packet. In the case of voice packets and data packets, each node would need to maintain two Routing tables; one focused on minimizing delay and the other on minimizing errors. This emphasis can be accomplished by setting the weighting coefficients appropriately. Because data packets have more stringent symbol error requirements, the forwarding protocol for data packets would have a larger weight applied to the *link-dependent resistance* ($\alpha_d > \beta_d$). Similarly, voice packets would have more emphasis on speed and would need to emphasize *node-dependent resistance*. ($\beta_d > \alpha_d$).

In [58], the authors continue to evolve the forwarding/routing protocol with an additional weighting based on energy expenditure associated with a path. The suggested energy expenditure cost function is simply a ratio of transmit power to minimum transmit power and an inverse ratio of code rate to maximum code rate. This evolution of the LRR routing protocol is only useful for a network that employs adaptive transmission. With the increasing popularity of adaptive communication channels, this type of improvement is worth more exploration.

In [59], the authors modify the LRR metric formulation to include an additive constant *c* that simply increases resistance associated with extra stops along the path. This is a simple method for deemphasizing paths that include extra stops.

As currently simulated, the MANET forwarding protocol only compares paths with zero or one intermediary stop. If the maximum distance between mobiles grows, it may become necessary to modify the protocol to consider paths with two intermediary stops.

g. Improved Modeling of Forward Error Correction

As discussed in Chapter III, the current MANET simulation only models a rough FEC process that checks a ratio of bit errors to total packet size. A higher-fidelity FEC approach will increase the accuracy of the model's performance estimates and could support investigations comparing different FEC implementations (i.e., convolutional codes vs. block codes, different code rates, etc.).

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A: PREDICTION-BASED SOLUTIONS TO THE URBAN COTM SIGNAL FADE

Another area of investigation into possible mitigations of the urban LMSC fade problem is fade prediction. This solution is inherently different and essentially unrelated to the Neighbor routing protocol introduced in Chapter III. The Neighbor protocol is a reactive mitigation strategy that circumvents link outages by routing traffic through an alternate path, and outage prediction is not as important or helpful. Without the Neighbor protocol, outage prediction could have a substantial impact on individual satellite link performance by avoiding transmission when successful transfer is unlikely or not possible. Instead, packet transmission could be timed to increase the probability of successful transfer and acknowledgement. Another approach would be for the mobile to predict an outage and notify the satellite before entering the shadow condition. If the satellite can be provided an approximate time of outage, it could also identify probable lost packets and prepare to retransmit them. If possible, ideal prediction would provide maximum system throughput for a single link. Two key benefits of a prediction-based solution are lack of dependence on nearby neighbors and a lack of impact on local ad-hoc network performance.

Terrestrial-to-satellite channel prediction is not a new area of investigation [8, 9]. In [9], Choi and Chan use a 2^{nd} order Autoregressive (AR) prediction filter to predict the channel attenuation due to scintillation and rain. The results are very promising for predicting one to three seconds into the future with prediction error variances less than 1 dB. We do not expect similar performance in applying this approach to the Urban COTM channel prediction. The variations in signal strength are substantially larger and much more abrupt. It is interesting to note that the prediction filter in [9] does tend to lag behind the actual instantaneous channel attenuation but not enough to impact the value of the overall approach. Unlike the urban building fade, atmospheric scintillation and rain are relatively slow-varying, well-behaved log-normal stochastic processes. In [8], Satorius and Ye go even farther. In addition to scintillation, they also examine channels

with multipath fading that can vary in signal strength by up to 30 dB. The signal strength transitions are not as abrupt as the COTM building fades but are certainly closer than the investigation found in [9]. To handle the more complex channel, Satorius and Ye implement a more complex predictor based on Subspace methods (i.e., MUSIC, ESPRIT) and reduce the predict-ahead time to 0.28 seconds. Satorius and Ye find prediction performance to be acceptable [8]. Neither of these works includes the use of a highly-directional antenna like that planned for the future TCA or the resulting urban building fade phenomenon.

We have investigated two approaches towards implementing a prediction-based solution. The first is discussed in Section A and focuses on using signal strength as measured by the high-gain SATCOM antenna discussed in Chapters I and II. The second approach is discussed in Section B and investigates the idea of using multi-path signal energy collected by an omni-directional antenna. It is important to note that initial investigations into prediction-based solutions are not promising, and these areas were not pursued to completion.

A. SIGNAL FADE PREDICTION WITHOUT MULTI-PATH

The most straightforward approach to predicting signal outages on the COTM link is to analyze the signal strength received by the COTM antenna and attempt to predict future signal strength.

For the direct prediction of LOS signal fade using the LOS signal itself, we used the empirical data presented and discussed in Chapter II. The initial goal is to predict one time step into the future. Because the empirical data was collected at a rate of 10 Hz, this one step prediction equates to 0.1 seconds, which is noticeably shorter than investigated by [8] and [9]. Multiple approaches were investigated including post-processing with autoregressive (AR) filters and real-time processing with linear adaptive filters. Each approach is discussed in more detail below.

An important facet of the prediction investigation is determining whether the prediction will be performed at the mobile or at the satellite. We again see that the path

delay makes the situation very different for each approach. The more valuable location for an accurate prediction process is at the satellite. The satellite is the node that will incorrectly transmit packets when the mobile will be blocked ~0.125 seconds in the future. While that kind of prediction would certainly be very valuable, it is also highly unlikely due to the long prediction time needed. The other option is for each mobile to predict its own link outages. As mentioned above, this could lead to a notification or warning message being sent to the satellite before a fade. The satellite could use that notification to delay transmittal and determine if any sent packets would need retransmission.

1. Autoregressive (AR) Filters: Post-Processing

Post-processing is not, in itself, a useful solution to this problem. If the satellite or a mobile is going to predict future signal strength, it is only relevant if it can be accomplished real-time and utilized as part of the routing protocol. But a post-processing solution can still be useful as a possible baseline for comparison. The empirical data presented in Chapter II was used to optimize AR prediction filters of varying filter length through application of the *covariance* and *modified covariance* methods [63]. Unlike the adaptive filters, a longer filter length does provide some small amount of improved performance. Figure 66 shows the prediction error variance σ_e^2 as a function of filter order for both methods. The first approach is the covariance method which relies solely on the forward prediction error and only moves the filter through the data in the direction of time advancing. This provides a prediction error variance of

$$\sigma_e^2 = \frac{S^f}{N}, \tag{5.3}$$

where S^f is the sum of the forward prediction errors squared, and
N is the size of the data set.

The modified covariance method is only slightly different in that it moves the filter through the data in both the forward and backward directions. The only impact to computing the prediction error variance is the replacement of the forward prediction errors squared S^f with the forward and backward prediction errors squared S^{fb}.

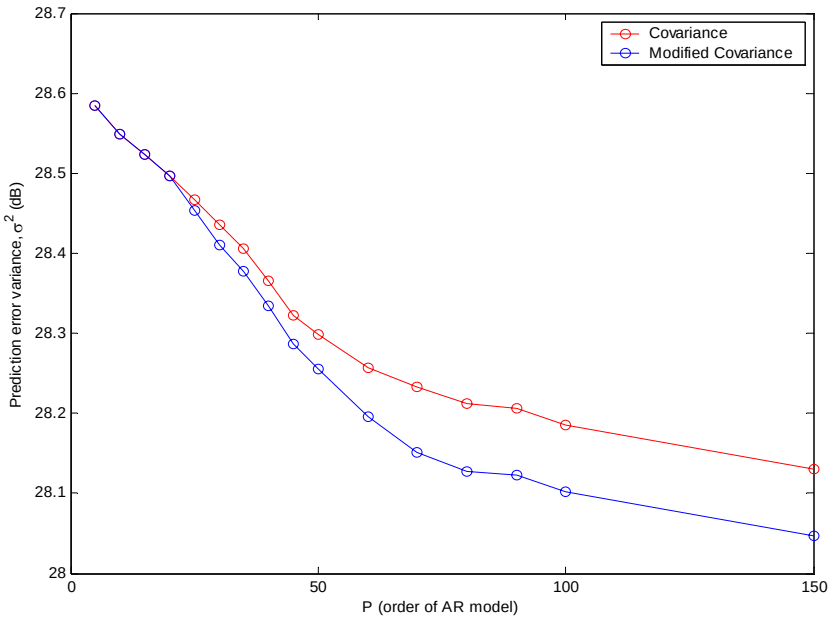

Figure 66. Theoretical prediction error variance as a function of filter order *P* using either the covariance or modified covariance methods.

The *y*-axis scale is stretched out to more clearly show the error decreasing with filter order but the rate of descent is very slow and appears to be approaching an asymptotic limit. It is not surprising that the modified covariance method provides a small improvement in performance. Because the modified covariance method minimizes the forward and backward prediction error, it essentially doubles the data set and is expected to provide better performance [63]. The benefits are quite small, however, and even at $P = 150$, the benefit is only ~0.1 dB. The actual signal strength compared to the predicted signal strength as a function of time is shown in Figure 67. The filter prediction output appears to lag and undershoot the true signal strength. The undershoot is not really important if we are only using the filter to predict a channel fade, but the time delay is problematic for this particular application.

Ultimately, the results of this approach are disappointing. The prediction error variance is very large compared to the actual signal and substantially larger than the results achieved against other channel fade problems (e.g., weather, scintillation,

multipath) [8, 9]. More importantly, the prediction of deep urban fades lags behind the actual event and does not provide timely side information to the protocol.

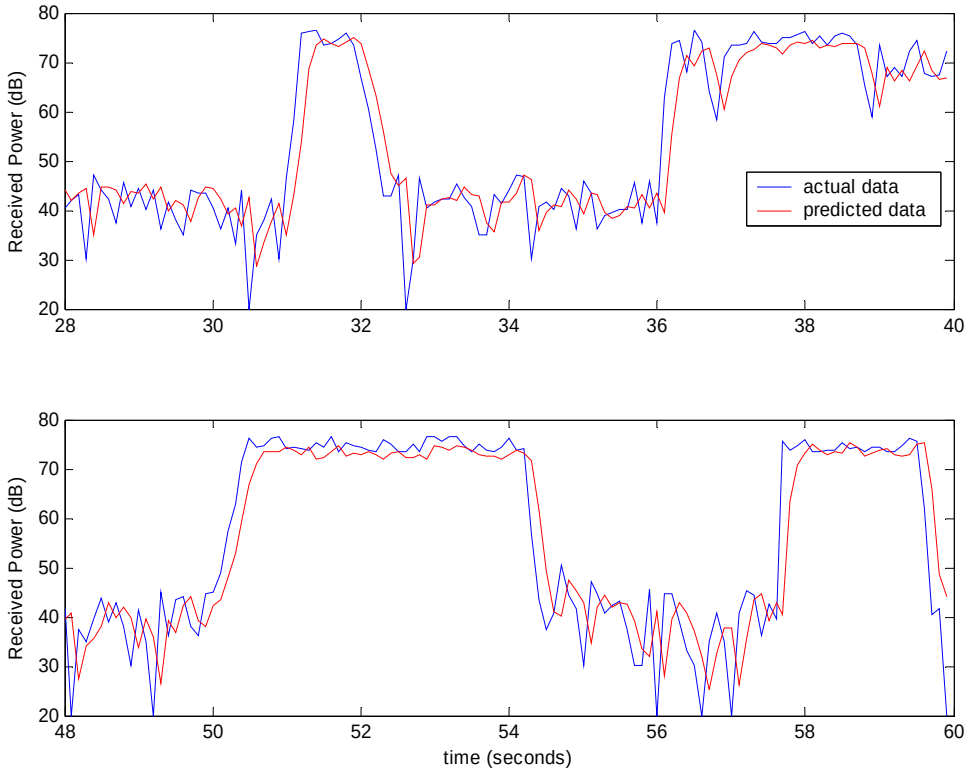

Figure 67. Input data vs. filter prediction for two separate time windows ($P = 100$).

2. Real-time Prediction Using Adaptive Filters

Each of the next three attempts to accurately predict the LOS signal are based on linear prediction by use of linear adaptive filters. They include Normalized Least Mean Squares (NLMS), Gradient Adaptive Lattice Ladder (GAL^2), and Recursive Least Squares (RLS).

a. Normalized Least Mean Squares

The first adaptive linear prediction filter attempted was the NLMS. A short sample of the filter output, for a filter of length $P = 5$, is found in Figure 68. It is plotted along with the empirical data and resulting error for direct comparison. Unlike the AR filter discussed previously, the NLMS performance improved as the filter order decreased. Like the other filters discussed in this chapter, the filter output lags the actual data. A small investigation into other filter lengths showed that the lag grew with filter order. Of special interest are the three state transitions (connection to fade) that occur in this small window at $t = 120.6$, 125.7, and 129.3 seconds. The fifth order NLMS also predicts state transitions, but not until $t = 120.8$, 125.9 and 129.4 seconds, respectively. Since the mobile has already entered the fade state by that point, the filter does not provide any benefit. This adaptive filter implementation does have less overshoot than the GAL (see Section b.) and actually seems to have a smoothing effect on the signal. The error variance is noticeably larger than that of the AR filter discussed earlier. The fifth order NLMS had a σ_e^2 of ~ 41 dB.

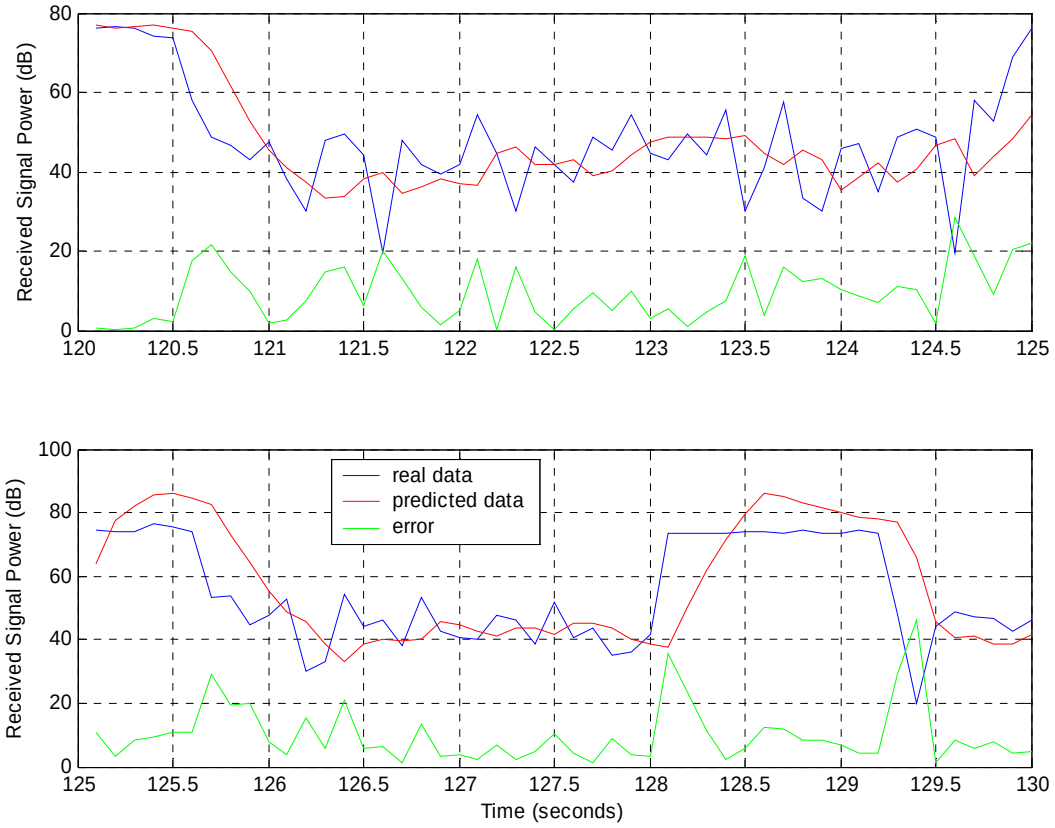

Figure 68. Output of NLMS prediction filter compared to truth. The difference is presented as error vs. time.

b. Gradient Adaptive Lattice Ladder

The results for the GAL[2] filter are shown in Figures 69 and 70. The filter performance as a function of filter order P and forgetting factor β_{GAL} is shown in Figure 69. A simple parametric study shows that the optimal β_{GAL} is between 0.7 and 0.8. Obviously, as the filter gets smaller, the importance of the forgetting factor is reduced since there are fewer stages in the filter and, therefore, fewer past data points to be considered. It is also obvious that the filter order has a large impact on performance, especially at non-optimal values for β_{GAL}. As the filter gets longer, it has more memory of past values, which for this situation is harmful to performance. When the signal

strength plummets from a strong connection to a deep fade, a large filter (e.g., P = 10) takes several time steps to clear its memory of the high signal strength. Meanwhile, those high signal strength values continue to hamper the filter's ability to predict the low signal strength state.

The prediction error performance of the GAL^2 filter is noticeably worse than the AR or the NLMS. For a five stage GAL^2, the prediction error variance is near 60 dB.

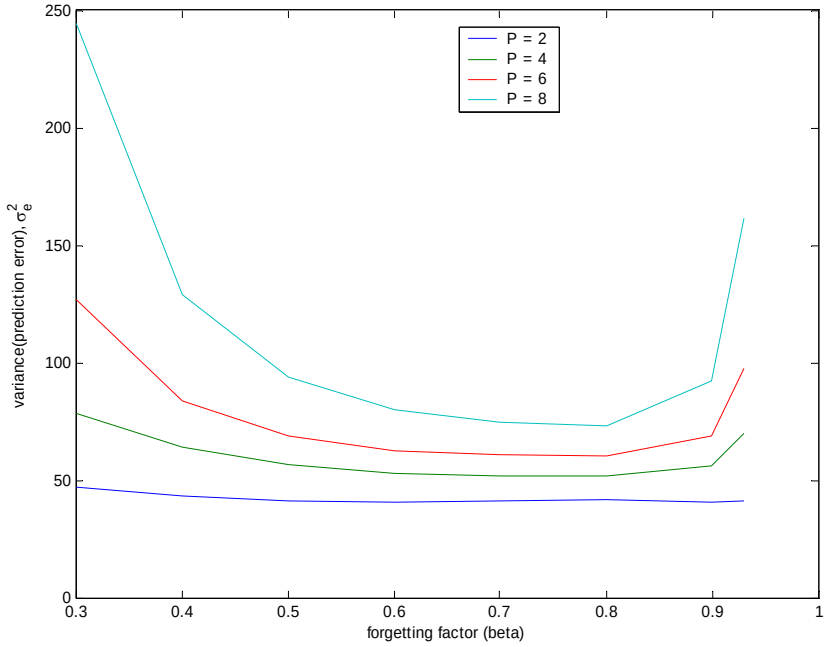

Figure 69. Filter performance σ_e^2 as a function of filter order P and forgetting factor β_{GAL}.

The predicted signal strength is compared to the actual data in Figure 70. The forgetting factor β_{GAL} is set to a favorable value of 0.8 and filter order is set to P = 5. While a shorter filter would perform slightly better, this allows better comparison with the NLMS and AR filters already presented. Two major results are apparent. The first, unexpected result is that the filter lags behind a precipitous change in signal strength

(e.g., connection/fade state change) and filter memory prevents accurate prediction. The second result is a strong tendency to significantly overshoot those same precipitous changes, again caused by filter memory.

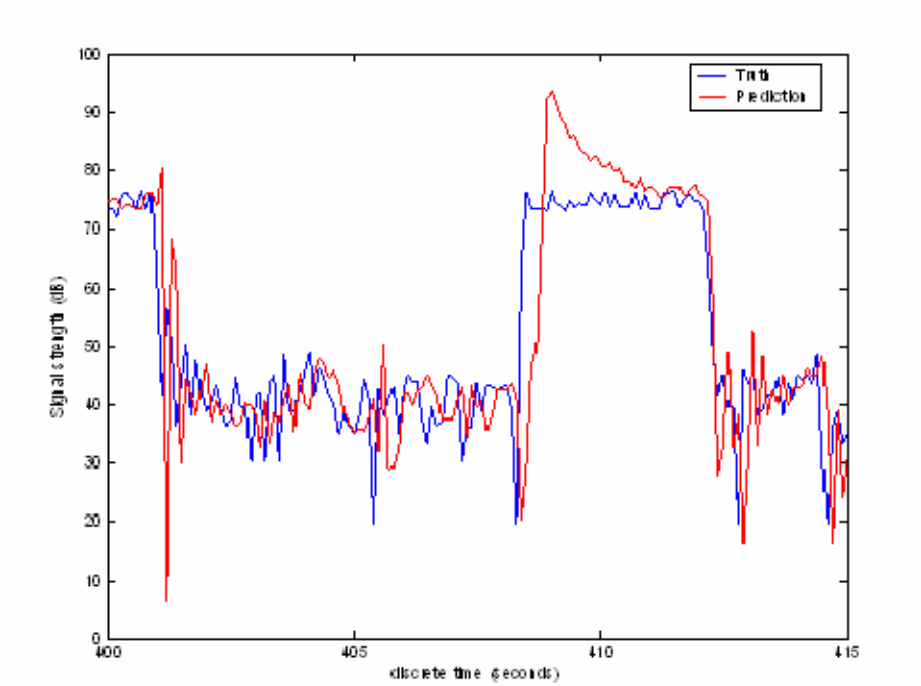

Figure 70. Input data vs. filter prediction ($P = 5$, $\beta_{GAL} = 0.8$).

c. Recursive Least Squares

The recursive least squares (RLS) filter has both the best and worst performance of any prediction method investigated. A 16 second comparison of filter output to the true signal strength for a filter of length $P = 5$ is shown in Figure 71. This particular snapshot was selected because it included two connection-to-fade state transitions and two fade-to-connection state transitions. The RLS approach performed the best with respect to occasionally predicting state transitions before the event occurred. Unfortunately, it also frequently predicts state transitions that do not actually occur. An example of this behavior can be found at $t = 125.5$ seconds. The RLS filter predicts a

transition to the fade state, but at t = 125.8 seconds, it also predicts an erroneous transition back to the connection state. Another example of this behavior can also be seen at t = 130.8 – 131.5 seconds. Separate from the ability to predict state transitions, the RLS filter has extremely poor performance at generally predicting signal strength with a prediction error variance σ_e^2 of over 700 dB. This is over an order-of-magnitude worse than any of the other filter options.

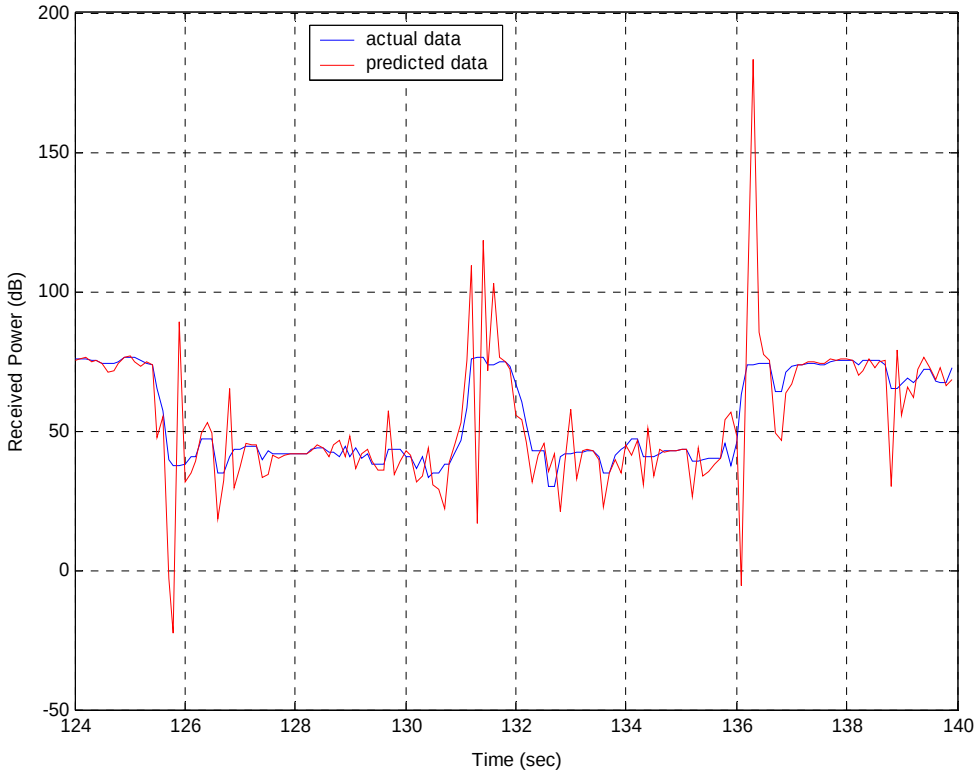

Figure 71. Output of RLS filter compared with actual signal strength (P = 5).

3. Summary

In the end, none of the adaptive prediction methods provided satisfactory performance for this particular application. Even the best mean squared error, from NLMS, was substantially larger than those errors seen in [8] and [9]. The second and more important problem was the inability to predict a state transition before it happened.

In [8] and [9], the authors experience a small lag in the prediction as compared to the actual channel performance, but for the channels and applications investigated, that lag was acceptable. For the application considered here, it is crucial that the predictor accurately foresee a state change before it occurs.

Poor prediction performance is not surprising. As mentioned earlier in this work, the empirical data was collected at a coarse temporal resolution of 10 Hz. The urban signal fade due to building blockage is a very abrupt process, but is made even more so with the low resolution data. As introduced in Chapter II, the data set is fairly well approximated as a Markovian chain. This relationship implies difficult prediction because a true Markovian process is inherently unpredictable due to its *memoryless* nature [27]. As the signal strength becomes more Markovian, it becomes harder to accurately predict. One solution to reducing the Markovian nature of the data set is to collect it at a higher frequency. Another possibility is to generate data at a higher temporal or spatial resolution with a model. This concept is explored more in the next section. Additional recommendations are provided in the Appendix A conclusion.

B. SIGNAL FADE PREDICTION WITH MULTI-PATH

Section A showed that direct prediction of the signal strength using only the LOS signal to be difficult with linear prediction filters. A major factor in this difficultly was the abrupt and precipitous nature of the change is signal strength associated with a connection/fade state transition. A possible mitigation to that aspect of the problem is to collect both LOS and multipath energy. The multipath energy does not change as abruptly and may provide some performance improvements. For this approach to be possible, the mobile user may need to employ an omni-directional antenna in addition to the highly directional COTM antenna discussed in Chapter I. If the mobile user is part of a local area ad-hoc network, then it will probably already have something approximating an omni-directional antenna.

Ray trace scatter modeling (RTSM) is a theoretical approach to analyzing this mitigation strategy. It is primarily based on the geometric optics theories of scattering,

reflection and diffraction. These theories are applicable for high frequency radio waves, especially when the scattering objects are significantly larger than a wavelength. This theoretical approach is important because it provides a deeper understanding of the problem and can fill in some missing information regarding variations in signal strength just before or after a deep fade. An introduction to the supporting theories of RTSM (i.e., geometric optics, diffraction) can be found in references [64, 65]. There are commercial software applications available that compute ray trace scattering. Urbana by Science Applications International Corporation (SAIC) is an example from the 1980s, but it unfortunately requires the transmitter and receiver to be much closer together than a mobile-to-satellite link can accommodate.

The empirical data discussed in Chapter II was collected at a rate of 10 Hz. As mentioned in Chapter II, at this rate the signal strength appears to have nearly instant state transitions between connections and fades. In many cases, the signal strength changes by 20 to 40 dB in a single time step. Other authors who have investigated the urban communications channel have collected test data at higher resolution [66, 67, 68]. While these references do not specifically discuss the Markovian or non-Markovian nature of the signal strength, they do provide general discussions that suggest less abrupt transitions in signal power. Based on the results presented in these articles, we propose that additional information may be available at higher resolutions and may be useful in fade prediction. While the best way to explore this possibility would be a repeat of the 2002 MOTM/COTM tests [5] with data collection rates over 100 Hz, a reasonable estimate can be acquired through RTSM.

To start this investigation, we developed a basic model that illustrated the first order impacts of passing from one building shadow to a connection state and back to another building shadow. The implementation approach of this model is discussed in the next section. Future work could involve a more complex model incorporating additional structures, especially behind the mobile, and modeling additional scattering.

1. Two-building Model

a. Model Introduction and Background

The purpose of this model was to focus closely on the two state transitions experienced by the urban communications link: 1) connection to deep fade and 2) deep fade to connection. A physical interpretation of the model is shown in Figure 72. The results were collected for small but equal steps in spatial coordinates. The results also reflect a constant time step if the mobile receiver has a constant velocity. As shown in the figure, the mobile receiver is in motion down the negative x-axis. Only two buildings are modeled for this focused analysis. The basic assumption of this model is that all urban areas have gaps of visibility to the satellite. The important moments in time are the transition from building shadow into visibility and then back into building shadow. From the definitions presented in Appendix B, these transitions involve the vehicle crossing the *incident field shadow boundary*. Building size and separation primarily effect the duration of connection or deep fade but are not expected to have a significant impact on the transitions themselves. In a general sense, the signal strength is cyclic going from connection to deep fade and back again. A model with two buildings provides one complete cycle in signal strength. Additional buildings would provide additional cycles in signal strength without adding any new information.

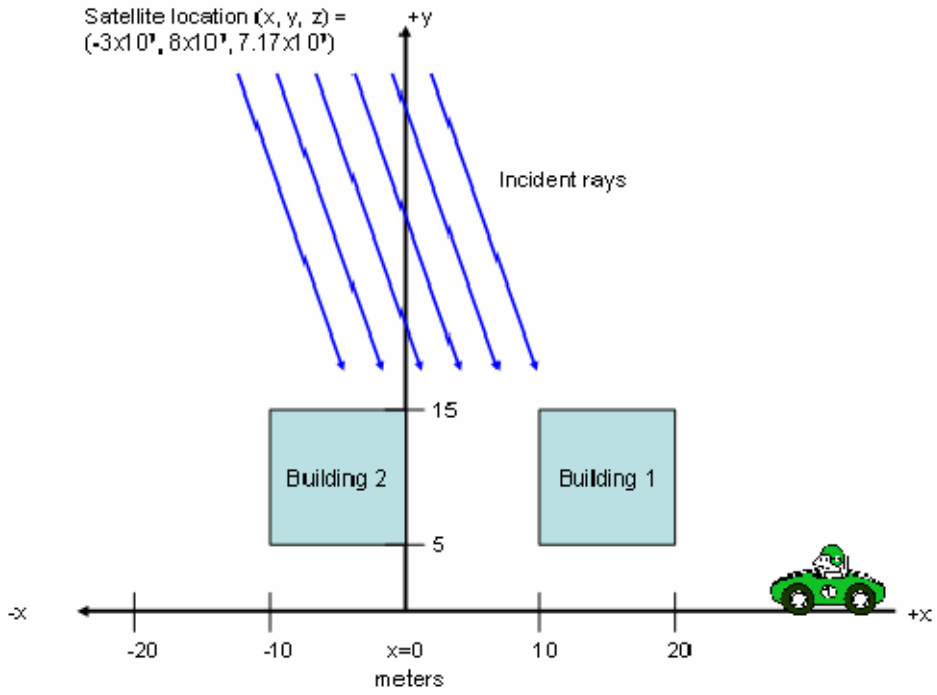

Figure 72. Overview of Two-building RTSM.

While building size and location does not have a significant impact on the transition from connection to fade, building shape and orientation may. Geometric optics theory is heavily influenced by the surface angle of the scattering object relative to the receiver. The surface orientation determines the direction of energy reflection and edge orientation determines the direction of energy diffraction. This specific model assumes that the buildings are in the traditional box-like shape with all right angles. It is also assumed that the building fronts are parallel to the vehicle motion as if the vehicle is driving down a city street. Additional building shapes and orientations are left for future work.

The goal of the simulation is to determine the amount of signal energy collected by the mobile antenna relative to the incident energy from the satellite. The simulation was completed for two different receive antennas: 1) omni-directional and 2) highly directive. The omni-directional antenna is modeled as an ideal isotropic antenna with no variation in antenna pattern. This provides a first order approximation of a dipole

or similar antenna that has a uniform pattern in azimuth. A dipole antenna does have significant pattern variation in elevation but that does not play a significant role in this problem. As the antenna moves along the *x*-axis, the direction of incoming energy changes primarily in azimuth. The highly directive antenna is modeled as a circular aperture but is a reasonable approximation for a parabolic reflector antenna.

The satellite has been arbitrarily located at $(x, y, z) = (-3\text{x}10^7, 8\text{x}10^7, 7\text{x}10^7)$ m. In reality, the location of the satellite relative to the mobile receiver is highly variable depending of the latitude, longitude and altitude of the mobile receiver and the longitude of the TSAT satellite in the geosynchronous belt. The arbitrary selection of the satellite location places the satellite at 45 degrees of elevation relative to the mobile receiver. Since the results are normalized to the LOS signal strength, the satellite's range along this vector has no impact on results. While the satellite's relative location has some impact on the general trends seen in the transition across an incident field boundary, they are fairly minor compared to the impacts on connection and fade duration. For the situation shown in Figure 72, the connection times are maximized if the satellite is located somewhere along the *y*-axis. The visibility window between building shadows is ten meters, but as the satellite shifts in either direction, the building shadows get wider and the visibility gap gets smaller. At some angle the visibility gap drops to zero, and the receiver only has reflected or diffracted energy available. For this specific model, the visibility gap disappears at an angle of 45 degrees off the *y*-axis.

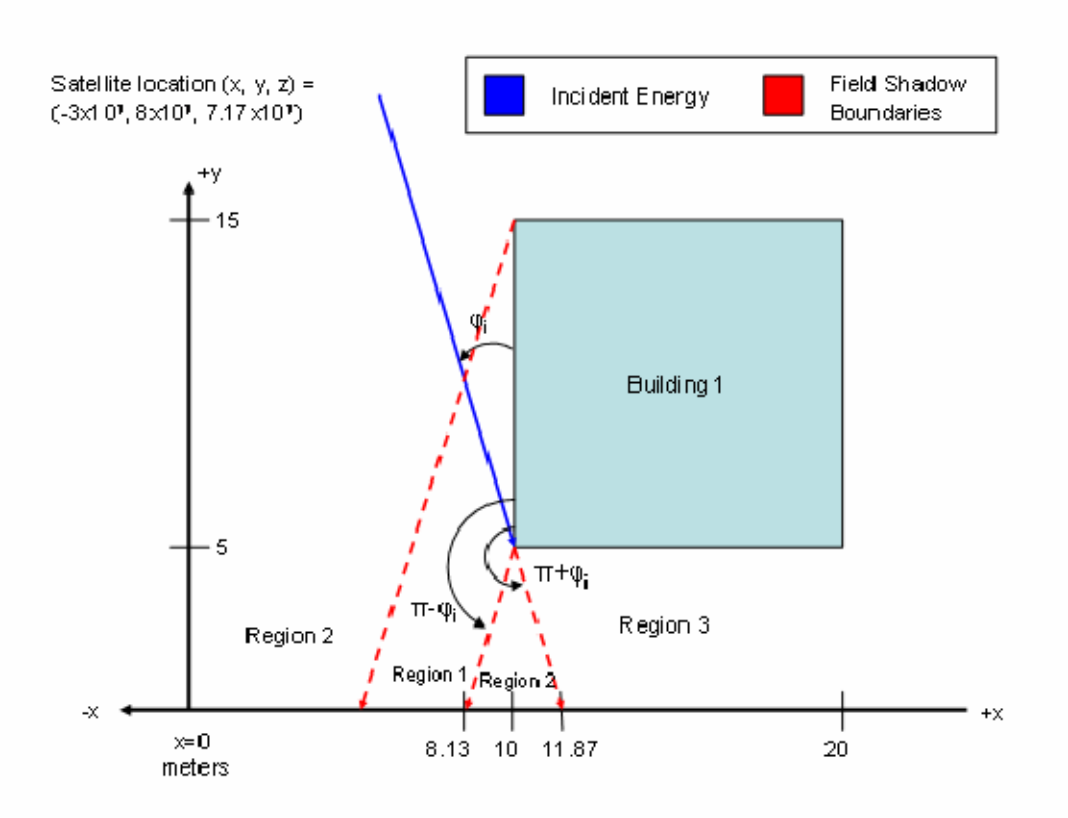

Figure 73. Close-up of Building 1.

Figures 73 and 74 show additional details regarding key model geometries. When focused on Building 1 as the scattering object, the $-x$ wall is the relevant reflecting surface and the $(-x, -y)$ corner is the relevant diffracting edge. While the $(-x, +y)$ edge will also cause diffraction, the vehicle will only see this input after it has left the building's shadow and is illuminated by incident energy. The additional input from the $(-x, +y)$ edge is negligible when compared to the incident field. The mobile vehicle starts in region three which is the building's shadow. The only energy available during this portion of the route is diffracted energy. When the mobile antenna reaches x = 11.87m, it passes through the incident field shadow boundary and the satellite becomes visible. It is in region two for 3.74 m and then passes through the reflected field shadow boundary. At this point, the mobile antenna is illuminated by diffracted, incident and reflected rays. The angle of incidence φ_i is 20.5 degrees.

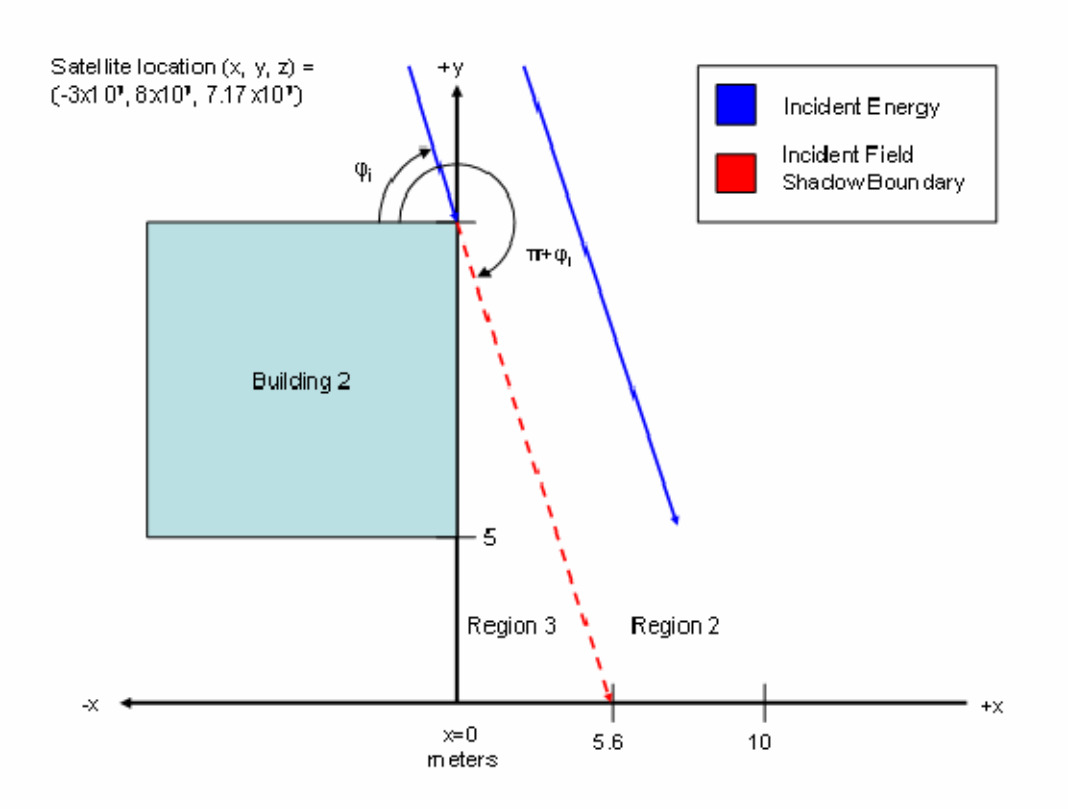

Figure 74. Close-up of Building 2.

Likewise, we can treat Building 2 as the next scattering object. The first significant difference is that Building 2 does not provide any reflected rays until the vehicle is past $x = -11.87$ m, which is well into the next connection state. From the perspective of this building, the vehicle will pass from region two to region three crossing the incident field shadow boundary. This is also the transition from the connection state to fade state. For Building 2, the angle of incidence is 69.5 degrees measured from the southern wall. If Building 2 were the only scattering object, the vehicle would start in region two (diffracted and incident energy) and transition to region three (diffracted energy) at $x = 5.6$ m. When the two buildings are considered simultaneously, there is only one modification required: Building 1 blocks the diffracted energy from Building 2 until the vehicle reaches $x = 15$ m and has clear visibility of Building 2.

b. Results

The results of the Two-building Model are shown in Figures 75 through 77. The first results are for a vehicle using an isotropic omni-directional antenna. For this case, Figure 75 separates the contributions of the geometric optics field, consisting only of incident and reflected signal energy and the diffracted field. These results assume that the buildings have an arbitrary reflection coefficient of 0.5. As the reflection coefficient grows, the total field also grows from $x = 0$ to 6 m, where the collected energy is all reflected by Building 1. The reflection coefficient also has an impact on the constructive / destructive interference pattern from $x = 6$ to 8 m. As the reflection coefficient grows, the constructive interference also grows to an ideal limit of twice the incident energy. Likewise, the destructive interference decreases towards an ideal limit of zero (perfect cancellation).

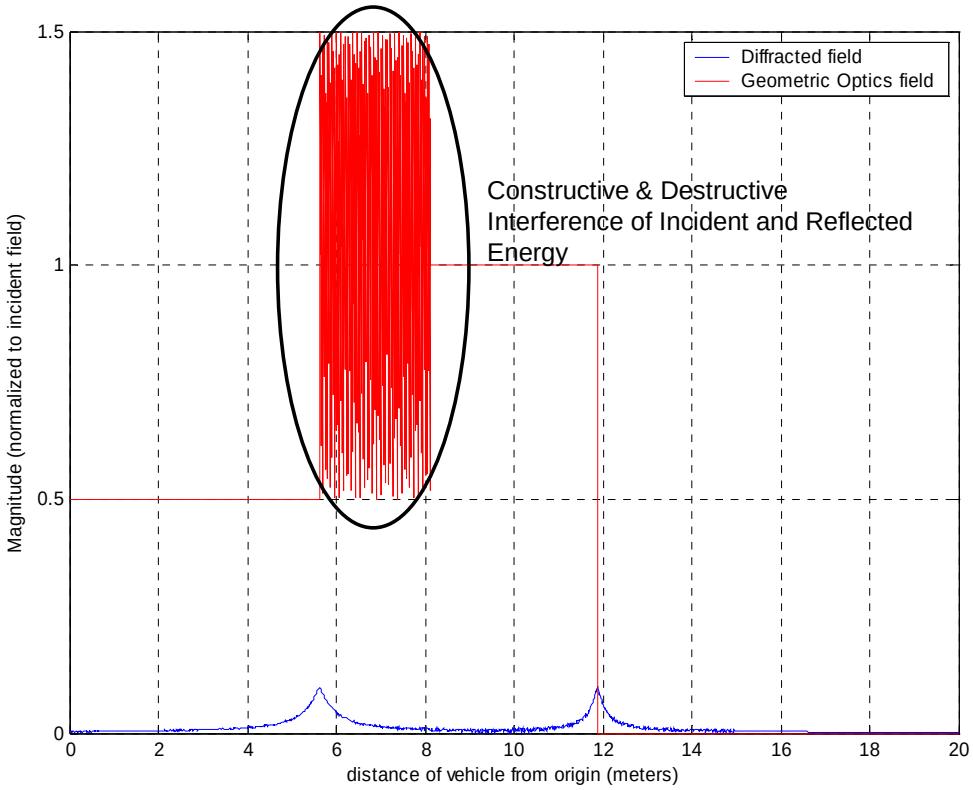

Figure 75. Separate geometric and diffracted fields as seen by an omni antenna (reflection coefficient = 0.5).

The diffracted field magnitude peaks at the incident field boundary with 10% of the incident (e.g., LOS) field strength, or −10 dB. The second incident field boundary, caused by Building 2, actually has a broader diffracted field. Figure 76 combines the incident, reflected and diffracted fields into a total electric field **E** which adds the constructive and destructive interference of the diffracted field to the geometric optics field.

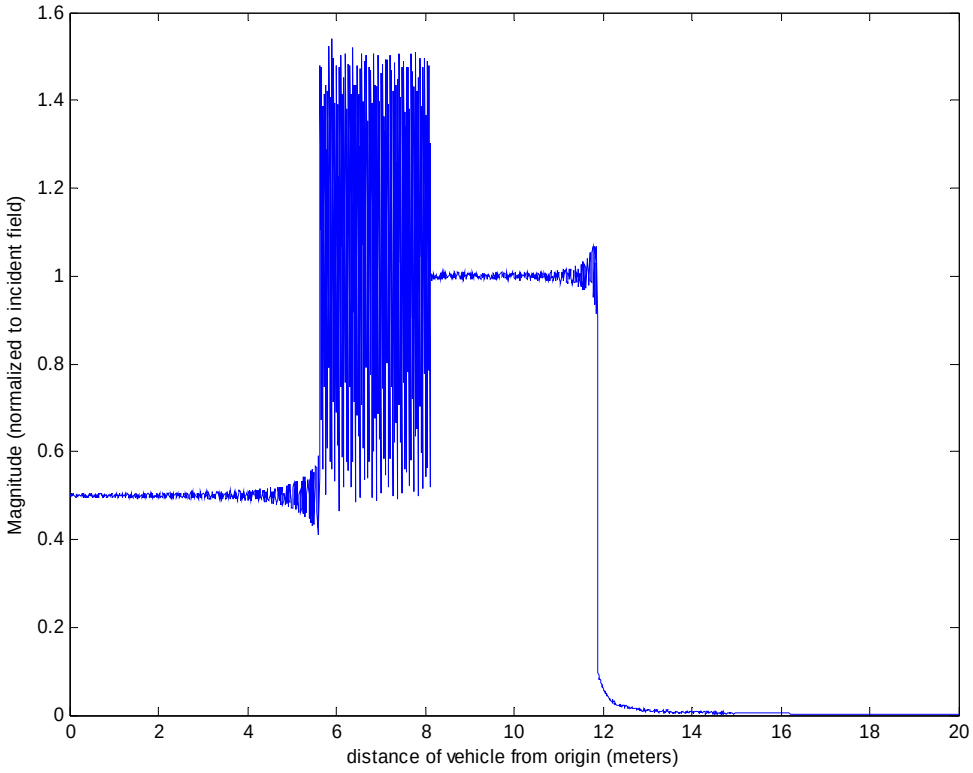

Figure 76. Total electric field as seen by an omni antenna (reflection coefficient = 0.5).

The most important facet to investigate is the transition from connection to fade that occurs near $x \sim 6$ m. It would appear that for this particular geometry, the diffracted energy that might indicate a coming field boundary has been washed out by the incident and reflected energy. If the vehicle changes direction and drives from $x = 0$ to $x = 20$ m along the $+x$ axis, then the reflected energy is not a factor and the diffracted field

provides a small but noticeable indication that a field boundary is approaching. Of course, this model does not include the various noise sources that are part of a real-world implementation. Those noise sources can be as large or larger in magnitude than the diffracted field.

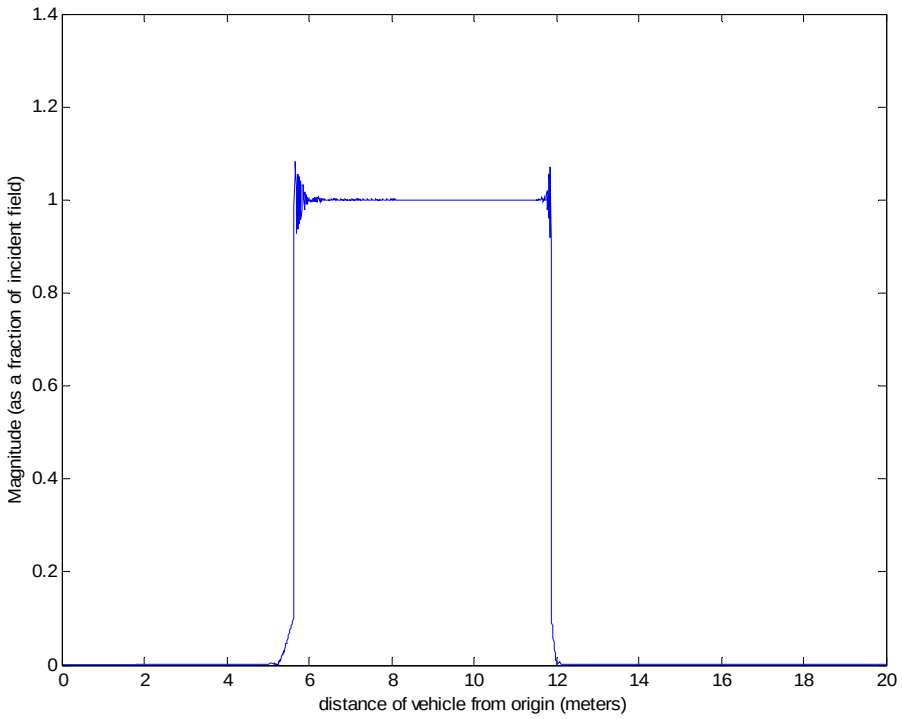

Figure 77. Total electric field as seen by a highly directional antenna

The second implementation of the Two-building Model replaced the mobile's omnidirectional antenna with a highly directional one to better match the COTM antenna discussed in previous chapters. Figure 77 presents the updated results for the total combined electric field magnitude. The directional antenna pattern clearly has a large impact of the energy collected by the transceiver. One impact is a large reduction in the collection of reflected energy which has a large angle off of antenna boresight.

The diffracted energy is still apparent but has smaller shoulders in the sense, that its magnitude is diminished as the mobile gets farther from the incident field boundary. This is again due to the antenna pattern.

The combination of these changes may surprisingly work in the favor of the highly directional antenna. The loss of diffracted energy away from the incident field boundary is an important impact and may prevent the diffracted signal from being lost in noise. On the other hand, the severe rejection of off axis energy could help reduce the input noise caused by multi-path energy and help the diffraction warning to stand out.

c. Summary for Prediction with Multipath

The Two-building Model represents a very idealized and limited investigation into the possibility of using diffracted energy as a warning of pending signal fade. Initial conclusions show that diffracted energy does provide an increase in the mean signal energy available to the receiver as the mobile approaches the incident field boundary (which marks the beginning of the fade state). The system noise floor and signal processing capabilities will determine how far before the fade the receiver is able to accurately detect the extra input energy. It is interesting to note that the directional system may provide even better performance at fade prediction because it removes a significant amount of reflected energy and minimizes problems associated with multi-path interference. The Appendix A conclusion includes recommendations for future work.

C. APPENDIX A CONCLUSIONS

Initial attempts at LOS-only fade prediction have shown poor performance. These difficulties are related to the abrupt transitions of signal strength that appear in a low resolution input signal. One possible mitigation to this problem is suggested by the "Two-building" ray trace scatter model, which suggests that prediction performance

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B: MATLAB IMPLEMENTATION NOTES AND LESSONS LEARNED

A. MATLAB IMPLEMENTATION

The research results presented in the dissertation are primarily based on Matlab routines. Central examples include the Baum-Welch parameter estimation found in Chapter II, the network simulations discussed in Chapters III and IV, and the linear prediction attempts presented in Appendix A.

The relevant Matlab m-files are available to the interested reader by contacting the author at tagillespie@gmail.com or Dr. Clark Robertson, Naval Postgraduate School, ECE Dept., at CRobertson@nps.edu.

B. LESSONS LEARNED

1. Dynamic Memory

For network simulation projects, it is important to understand that Matlab is not a compiled language and uses dynamic memory allocation. The Mathworks website provides a straightforward explanation of the dynamic memory process:

"*When you assign any type of data ... to a variable, MATLAB allocates a contiguous block of memory and stores the array data in that block. It also stores information about the array data, such as its data type and dimensions, in a separate, small block of memory called a header. The variable that you assign this data to is actually a* pointer *to the data; it does not* contain *the data.*

If you add new elements to an existing array, MATLAB expands the existing array in memory in a way that keeps its storage contiguous. This might require finding a new block of memory large enough to hold the expanded array, and then copying the contents of the array from its original location to the new block in memory, adding the new elements to the array in this block, and freeing up the original array location in memory. [69]"

This aspect of Matlab is especially important in network simulation because of the frequent queue manipulations that are central to the process (i.e., adding or deleting packets in various queues). The original version of the Model 1 simulation established a

matrix to represent each queue. When a new packet was generated or received, the matrix size increased by one row. When a packet was transferred or deleted, the top row of the matrix was deleted. As explained by Mathworks, this approach is read and write intensive, which is slow, especially when the matrices are large. The final (revised) version of the simulation is modified to preallocate memory for large matrices (e.g., queues). The goal of memory preallocation is to keep the queue *matrix* fixed in size but allow the packet queue to grow or shrink depending on packet traffic. To achieve this goal, an *active* queue is defined inside the fixed matrix where active is defined to include packets that have yet to be serviced. The active queue is identified by establishing read and write pointers to represent the top and bottom of the queue, respectively. The remainder of the fixed matrix contains inactive (already serviced) packets that can be ignored.

The memory preallocation approach has both positive and negative impacts on the simulation. One negative impact is the substantial increase in simulation complexity involved with manipulation of the pointers. When a pointer reaches the bottom of the preallocated matrix, it needs to be cycled back to the top of the matrix. This means that sometimes the active queue wraps around the bottom of the preallocated matrix and continues at the top. Packet management processes need to watch for this situation and only analyze the appropriate active queue contents. Another example of increased complexity is the difficulty of removing a packet or row from the center of the active queue without changing the size of the preallocated memory. An action that would traditionally take a single command now takes 64. A second negative impact of the preallocation approach is the increased use of computer memory. Instead of maintaining only the active packets in memory, the preallocated matrix includes some number of stale or serviced packets that are also maintained in memory. One way to minimize this impact is to appropriately size the preallocated memory. This is discussed later in this section. Memory management issues are not entirely attributable to memory preallocation and the topic is discussed more in Section B. Fortunately, memory preallocation also has an important positive impact on the simulation: a substantial savings in simulation runtime. Use of the ***tic*** and ***toc*** commands allows comparison of

run times. Results vary with simulation specifics, but runtime savings are frequently over 75% when comparing the original simulation with the revised preallocated version.

An important part of preallocating queue memory is selecting the correct matrix size. If the matrix is too small, the active queue will grow beyond the size of the matrix, and the pointers will make a complete cycle through the queue and start writing over active packets. If the matrix is too large, then the simulation may become memory limited, which leads to a constraint on simulated time duration. Obviously, the optimal queue size is dependent on simulation input parameters. The two parameters that are most important are packet generation rate and average fade duration. The packet generation rate directly influences the instantaneous number of packets appearing in the network. Fade durations cause packets to linger in queues longer, leading to the possibility of larger maximum active queue size. Fade duration statistics are related to Markov transition probabilities out of the fade state which is annotated as a_{01} for this discussion (see Section I.C for background on Markov models). If we combine the impacts of both parameters, we can combine the directly proportional impact of packet generation rate and the inversely proportional impact of the fade-to-connection transition probability:

$$Buffer \propto \frac{k\, r_p}{a_{01}}, \tag{5.4}$$

where k = scaling constant,
r_p = packet generation rate, and
a_{01} = probability of transitioning from state 0 (fade) to state 1 (connection).

2. Memory Management and Variable Data Type

As mentioned earlier in this section and in Chapter IV, memory shortages have a direct impact on several simulation constraints including the number of mobiles and time-steps simulated. As mentioned in Section A, the problem is exacerbated by preallocating memory to queue matrices, but the benefits in runtime are worth the expense. One way to mitigate this problem is to reclassify of simulation variables as either *integer* or *single* data types. The Matlab default data type for all variables is double, which requires eight

bytes per element. By reclassifying a variable to the *single* data type, we can achieve a 50% savings and store the variable with only four bytes of memory. Similarly, variables that can be reclassified as integers save 87.5% in memory usage.

3. Language Selection

As discussed in Sections A and B, the network simulations require a large amount of memory and have long run-times. For a personal computer (PC) with Windows XP, Matlab 7.0 (Release 14), 3 GB of RAM and a 3.0 MHz central processor, the run-time is approximately 19 hours for 20 iterations of 400,000 time steps each. The network simulation does not currently include a significant amount of matrix operations and does not take advantage of Matlab's strengths. Instead, the simulation does involve many ***for*** loops and ***if-then*** statements. Run-times could be substantially reduced by re-implementing the simulation in a language that is closer to machine-language and does not use dynamic memory (e.g., Fortran, C, etc.).

LIST OF REFERENCES[70]

[1] D. A. Fritz, B. T. Doshi, A. C. Oak, S. D. Jone, J. L. Burbank, H. L. Miller, J. D. Oetting, R. M. Collins, L. A. Gonzalez, and R.A. Nichols. "Military Satellite Communications: Space-Based Communications for the Global Information Grid." *John Hopkins APL Technical Digest*, vol. 27, no. 1. pp. 32- 40. 2006.

[2] James W. Canan, "Timing in battle: The T-Sat edge," Aerospace America, pp. 39-43, Jan. 2006.

[3] Ben Iannotta, "Playing if Safe; TSAT puts a premium on tried-and-tested technologies." *C4ISR The Journal of Net-centric Warfare*, vol. 6, no. 7, pp. 26-29, Aug 2007.

[4] Personal communications with Dr. Scott Stadler and Dr. Jeff Schodorf, MIT Lincoln Laboratory, Massachusetts.

[5] J. B. Schodorf, *EHF Satellite Communications on the Move: Experimental Results*, Technical Report 1087, Lincoln Laboratory MIT, Boston, MA, Aug 2003.

[6] J. F. Hayes, "Adaptive Feedback Communications," *IEEE Trans. Commun. Technol.*, vol. COM-16, pp. 29-34, Feb. 1968.

[7] James K. Cavers, "Variable-Rate Transmission for Rayleigh Fading Channels," *IEEE Trans. Commun.*, vol. COM-20, pp. 15-22, Feb. 1972.

[8] E. H. Satorius and Zhong Ye, "Adaptive Modulation and Coding Techniques in MUOS Fading/Scintillation Environments," *IEEE Military Communications Conference Record*, pp. 321-327, Oct. 2002.

[9] J. P. Choi and V.W.S. Chan, "Predicting and Adapting Satellite Channels with Weather-Induced Impairments," *IEEE Trans. Aerospace and Elec. Sys.*, vol. 38, no. 3, pp. 779-789, July 2002.

[10] [Online] "Communications-on-the-Move Antenna," www.l-3com.com/products-services/productservice.aspx?ID=125, Oct 2007.

[11] [Online] "45 cm Communication on the Move, In Motion, Antenna System," http://www.perlagrp.com/php/displayProducts.php?catID=23, Oct 2007.

[12] R. H. Clarke, "A Statistical Theory of Mobile-Radio Reception," *Bell Syst. Tech. J.*, vol. 47, pp. 957-1000, July-Aug. 1968.

[13] J. F. Ossana, "A Model for Mobile Radio Fading Due to Building Reflections: Theoretical and Experimental Fading Waveform Power Spectra," *Bell Syst. Tech. J.*, vol. 43, no. 6, pp. 2935-2971, Nov 1964.

[14] G. L. Turin, et al., "A Statistical Model of Urban Multipath Propagation," *IEEE Trans. On Veh. Technol.*, vol. VT-21, no. 1, pp. 1-8, Feb 1972.

[15] F. I. Meno, "Mobile Radio Fading in Scandinavian Terrain," *IEEE Trans. on Veh. Technol.*, vol. VT-26, no. 4, pp. 332-335, Nov. 1977.

[16] H. Suzuki, "A Statistical Model for Urban Radio Propagation," *IEEE Trans. Commun.*, vol. COM-25, no. 7, pp. 673-680, July 1977.

[17] F. Hansen and F. I. Meno, "Mobile Fading- Rayleigh and Lognormal Superimposed," *IEEE Trans. on Veh. Technol.*, vol. VT-26, no. 4, pp. 332-335, Nov 1977.

[18] Chun Loo, "A Statistical Model for a Land Mobile Satellite Link," *IEEE Trans. on Veh. Technol.*, vol. VT-34, no. 3, pp. 122-127, 1985.

[19] E. Lutz, D. Cygan, M. Dippold, F. Dolainsky and W. Papke, "The Land Mobile Satellite Communication Channel – Recording, Statistics, and Channel Model," *IEEE Trans. on Veh. Technol.*, vol. 40, no. 2, pp. 375-386, May 1991.

[20] G. E. Corazzo and F. Vatalaro, "A Statistical Model for Land Mobile Satellite Channels and its Application to Nongeostationary Orbit Systems," *IEEE Trans. on Veh. Technol.*, pp. 738-742, Aug 1994.

[21] A. Abdi, W. C. Lau, M. S. Alouini, M. Kaveh, "A new simple model for land mobile satellite channels," *Proc. IEEE Int. Conf. Communications 2001*, pp. 2630-2634, 2001.

[22] A. Abdi, W. C. Lau, M. S. Alouini, M. Kaveh, "A New Simple Model for Land Mobile Satellite Channels: First- and Second- Order Statistics," *IEEE Trans. Wireless Commun.*, vol. 2, no. 3, pp. 519-528, May 2003.

[23] B. Vucetic and J. Du, "Channel modeling and simulation in satellite mobile communication systems," *IEEE J. Select. Areas Comm.*, pp. 1209-1218, Oct 1992.

[24] Fernando Perez Fontan, Maryan Vazquez-Castro, Cristina Enjamio Cabado, Jorge Pita Garcia, Erwin Kubista, "Statistical modeling of the LMS channel," *IEEE Trans. on Veh. Technol.*, pp. 1549-1567, Nov 2001.

[25] W. Turin, *Digital Transmission Systems: Performance Analysis and Modeling*, McGraw-Hill, New York, 1999.

[26] Matthew Spencer, "Sensitivity analysis of Markov models for communities of competing sessile organisms" *Journal of Animal Ecology*, vol. 75, Issue 4, pp. 1024–1033, July 2006.

[27] Donald Gross and Carl Harris, *Fundamentals of Queueing Theory*, 3rd ed., John Wiley & Sons, Inc., New York, 1998.

[28] C. Alasseur, L. Husson and F. Perez-Fontan, "Simulation of rain events time series with Markov models," *Personal, Indoor and Mobile Radio Communications, 2004, 15th IEEE International Symposium on.*, vol. 4, Issue 5-8, pp. 2801–2805, Sep 2004.

[29] Ruey Tsay, *Analysis of Financial Time Series*, 2nd ed., John Wiley & Sons, Inc. Hoboken, NJ., 2005.

[30] L. R. Rabiner and B. H. Juang, "An Introduction to Hidden Markov Models," *IEEE ASSP Magazine*, pp. 4-16, Jan 1986.

[31] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257-286, Feb 1989.

[32] E. N. Gilbert, "Capacity of a Burst-Noise Channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253-1266, Sep 1960.

[33] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, pp. 1977-1997, Sep 1963.

[34] M. Mushkin and I. Bar-David, "Capacity and Coding for the Gilbert-Elliott Channels," *IEEE Trans. Inform. Theory*, vol. 35, no. 6, pp. 1277-1290, Nov 1989.

[35] G. Sharma, A. A. Hassan, A. Dholakia, "Performance Evaluation of Burst-Error-Correcting Codes on a Gilbert-Elliott Channel," *IEEE Trans. Commun.*, vol. 46, no. 7, July 1998.

[36] A. Goldsmith and P. Varaiya, "Capacity, Mutual Information, and Coding for Finite-State Markov Channels," *IEEE Trans. Inform. Theory*, vol. 42, no. 3, pp. 868-886, May 1996.

[37] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Statistic.*, vol. 37, pp. 1554-1563, 1966.

[38] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," *Ann. Math. Statistic.*, vol. 41, pp. 164-171, 1970.

[39] H. S. Wang and N. Moayeri, "Finite-state Markov channel—A useful model for radio communication channels," *IEEE Trans. on Veh. Technol.*, vol. 44, pp. 163-171, Feb 1995.

[40] Q. Zhang and S. A. Kassam, "Finite-state Markov model for Rayleigh Fading Channels," *IEEE Trans. Commun.*, vol. 47, no. 11, pp. 1688-1692, Nov 1999.

[41] H. Wakana, "A Propagation Model for Land-Mobile Satellite Communication," presented at the 1991 North American Radio Science Meeting and IEEE/APS Symposium, The University of Western Ontario, London, Ontario, Canada, Jun 1991.

[42] S. Ohmori, H. Wakana, and K. Seiichiro, *Mobile Satellite Communications*, Artech House, Boston, MA, 1998.

[43] Fulvio Babich, and Giancarlo Lombardi, "A Markov Model for the Mobile Propagation Channel," *IEEE Trans. on Veh. Technol.*, vol. 49, no. 1, pp. 63-73, Jan 2000.

[44] Jeff A. Bilmes, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," Technical Report no. TR-97-021. International Computer Science Institute. Berkeley, CA. Apr 1998.

[45] L. E. Baum and J. A. Egon, "An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology," *Bulletin of American Meteorology Society*, vol. 73, pp. 360-363, 1967.

[46] L.A. Liporace, "Maximum Likelihood Estimation for Multivariate Observations of Markov Sources," *IEEE Trans. Information Theory*, vol. IT-28, no. 5, pp. 729-734, Sep 1982.

[47] [Online] Kevin Murphy, Bayes Net Toolbox, http://bnt.sourceforge.net/ or http://www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html, Oct 2007

[48] William Stallings, *Data and Computer Communications*, 7th ed., Pearson Education Inc., Upper Saddle River, New Jersey, 2004.

[49] William Stallings, *Wireless Communications & Networks*, 2nd ed., Pearson Prentice Hall, Upper Saddle River, NJ, 2005.

[50] John G. Proakis, *Digital Communications*, 4th ed., McGraw-Hill, New York, 2001.

[51] Bernard Sklar, *Digital Communications, Fundamentals and Applications*, 2nd ed. Prentice Hall PTR, Upper Saddle River, New Jersey, 2001.

[52] [Online] "Mobile Ad-hoc Network," http://en.wikipedia.org/wiki/Mobile_ad-hoc_network, Oct 2007.

[53] M. B. Pursley, and H. B. Russell, "Adaptive Forwarding in FHSS Packet Radio Networks with Partial-Band Jamming," *IEEE Trans. On Communications*, vol. 41, no. 4, pp. 613-620, Apr 1993.

[54] M. B. Pursley and H. B. Russell, "Network Protocols for FH Packet Radios with Decoder Side Information," *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 4, pp. 612-620, May 1994.

[55] M. B. Pursley, H. B. Russell and P. E. Staples, "Routing Multimedia Packets in a Frequency-Hop Packet Radio Network," *IEEE Military Communications Conference Record*, pp. 220-224, Oct 1996.

[56] M. B. Pursley, H. B. Russell and J. S. Wysocarski, "Metrics for Characterizing Radio Links in a Frequency-Hop Packet Radio Network," *IEEE Military Communications Conference Record*, pp. 356-360, Nov 1997.

[57] M. B. Pursley, H. B. Russell, and J. S. Wysocarski, "Tradeoffs in the design of routing metrics for frequency-hop wireless networks," *IEEE Military Communications Conference Record*, vol. 1, pp. 65-69, Oct 2000.

[58] M. B. Pursley, H. B. Russell, and J. S. Wysocarski, "Energy-Efficient Routing in FH Networks with Adaptive Transmission," *IEEE Military Communications Conference Record*, pp. 1409-1413, 1999.

[59] M. B. Pursley, H. B. Russell, and J. S. Wysocarski, "Tradeoffs in the design of routing metrics for frequency-hop wireless networks," *IEEE Military Communications Conference Record*, vol. 1, pp. 65-69, Oct 2000.

[60] Ozan K. Tonguz, Gianluigi Ferrari. *Ad Hoc Wireless Networks: A Communication-Theoretic Perspective*, John Wiley & Sons, New York, May 2006.

[61] M. Mauve, J. Widmer and H. Hartenstein, "A Survey on Position-Based Routing in Mobile Ad-Hoc Networks," *IEEE Network*, vol. 1**,** no. 6, pp. 30-39, Dec 2001.

[62] Leland Blank, *Statistical Procedures for Engineering, Management and Science*, McGraw-Hill, New York, 1980.

[63] C. W. Therrien, *Discrete Random Signals and Statistical Signal Processing*, Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[64] J. D. Kraus and D. Fleisch, *Electromagnetics*, McGraw-Hill, New York, 1984.

[65] W. L. Stutzman and G. A. Thiele, *Antenna Theory and Design*, John Wiley & Sons, Inc., Hoboken, New Jersey, 1981.

[66] A. J. Rustako, N. Amitay, R. S. Owens, R. S. Roman, "Radio Propagation Measurements at Microwave Frequencies For Microcellular Mobile and Personal Communications," *IEEE International Conference on Communications, 1989 (ICC 89), Conference Record*, pp. 482-486. Jun 1989.

[67] A. J. Rustako, M. J. Gans, G. J. Owens, and R. S. Roman, "Attentuation and Diffraction Effects from Truck Blockage of a 11-GHz Line-of-Sight Microcellular Mobile Radio Path," *IEEE Trans. on Veh. Technol.*, vol. 40, no. 1, pp. 211-215, Feb 1991.

[68] E. Kubista, F. Perez-Fontan, M. A. V. Castro, S. Buonomo, B. R. Arbesser-Rastburg, and P. Baptista. "*Ka*-Band Propagation Measurements and Statistics for Land Mobile Satellite Applications," *IEEE Trans. on Veh. Technol.*, vol. 49, no. 3, pp. 973-983, May 2000.

[69] [Online] "Memory Allocation in MATLAB" http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/matlab_prog/f8-774052.html, Oct 2007.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Chairman
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

4. Prof. Clark Robertson
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

5. Prof. Hershel Loomis
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

6. Prof. Monique P. Fargues
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

7. Prof. Roberto Cristi
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

8. Prof. Brij Agrawal
   Department of Mechanical and Astronautical Engineering
   Naval Postgraduate School
   Monterey, California

9. Major Tim Gillespie
   United States Air Force
   Reston, Virginia